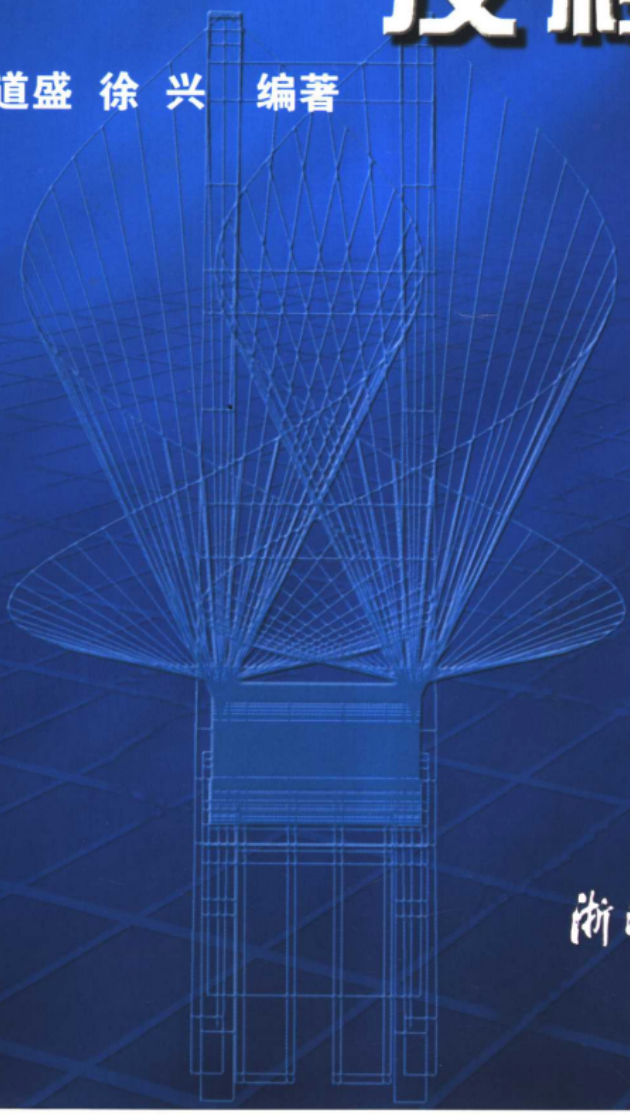
新世纪高等院校精品教材

FEIXIANXING YOUXIANYUAN JI CHENGXU

非线性有限元及程序

凌道盛 徐 兴 编著



浙江大学出版社

新世纪高等院校精品教材

ISBN 7-308-03872-6

9 787308 038720 >

ISBN 7-308-03872-6/TP・269　定价：36.00元

非线性有限元及程序

凌道盛　徐　兴　编著

浙江大學出版社

内容简介

本书除简要叙述线性有限单元法以外，重点阐述非线性有限单元法的基本理论。主要内容包括：非线性有限单元法的基础知识、主要算法、软件工程的基本思想、非线性有限元程序 V-FEAP 和工程算例；本书还包括作者的最新研究成果：改进等参元和实体退化单元。

本书通过介绍一个典型的非线性有限元程序 VFEAP，阐明非线性有限元程序的基本结构、算法的程序实现等问题，为读者在非线性有限单元理论和实际应用之间架起一座桥梁。

本书可作为土木、机械、力学等学科的研究生和高年级本科生的教材，也可供工程技术人员学习非线性有限单元法和程序时参考。

图书在版编目（CIP）数据

非线性有限元及程序 / 凌道盛，徐兴编著. —杭州：浙江大学出版社，2004.8

ISBN 7-308-03872-6

Ⅰ. 非… Ⅱ. ①凌… ②徐… Ⅲ. 非线性—有限元—程序设计 Ⅳ. TP311

中国版本图书馆 CIP 数据核字（2004）第 088330 号

责任编辑：杜希武 沈建国

封面设计：宋纪浔

出版发行：浙江大学出版社

（杭州浙大路 38 号 邮政编码：310027）

（E-mail: zupress@mail.hz.zj.cn）

（网址：http://www.zjupress.com）

排　　版：浙江大学电脑排版中心

印　　刷：德清第二印刷厂

印　　张：25

开　　本：787mm × 1092mm

字　　数：640 千

版 印 次：2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷

印　　数：0001–2000

书　　号：ISBN 7-308-03872-6/TP · 269

定　　价：36.00 元

前言

有限单元法是一种方法可靠、适用面很广的数值分析方法，其中线性的有限单元分析由于理论简明、应用简便，已被广大工程技术人员接受和应用。随着人类对客观事物认识的不断深入，设计水平的不断提高，基于线性理论的分析方法已不能满足理论研究和工程设计的需要，因此非线性有限元分析越来越得到广大科技工作者的重视。遗憾的是，非线性有限单元法所要求的基础理论知识更深、更广，其程序实现也更困难，而非线性问题本身的复杂性、多样性又决定了不存在一个能解决大部分（更不要说全部）非线性问题的通用非线性有限元分析软件。因此，非线性有限元分析要求使用者不仅要熟练地掌握有限单元的理论和技术，深刻理解研究对象的规律，并且要具有发展算法、扩充程序性能的能力，以便更有效、更经济地解决工程中的非线性问题。

从 1983 年开始，作者为浙江大学力学系和土木工程系的硕士研究生开设“有限单元法和程序设计”课程。随着有限元技术的发展，课程内容得到不断更新和丰富。1988 年开始开设“非线性有限单元法及程序设计”课程，讲稿几经修改，逐步形成本教材。该课程已成为土木工程博士学位课程。作者 20 年的教学经验表明，学习本课程的研究生开始的时候多少都会感到一些困难，但学完之后都会觉得有较大的收获，所学内容对其学位论文的研究帮助颇大。

本书重点阐述了非线性有限单元法的基本理论和程序设计，主要内容包括：非线性有限单元法的基础知识、主要的数值算法、软件设计工程的基本思想、非线性有限元程序 VFEAP 以及一些工程实例。为确保教材的系统性和可读性，本书还简要地叙述了线性有限单元法，同时收集了作者近年来的最新研究成果。通过系统地介绍一个典型的非线性有限元程序 VFEAP，阐明非线性有限元程序的基本结构、算法的程序实现等关键问题，为读者在非线性有限单元理论和实际应用之间架起一座桥梁。

本书第 1 章简要地介绍了有限单元法的发展历史和固体力学非线性问题的分类。第 2 章介绍非线性有限元分析过程中涉及的主要数值方法，包括线性与非线性代数方程组的解法、广义特征值问题的解法、瞬态响应问题的解法以及常用的各种数值积分公式等。这一章是非线性有限单元法的数学基础。

第 3 章介绍了笛卡儿坐标系下物体有限运动和变形的描述方法。第 4 章阐述了有限变形条件下的应力描述和微、积分形式的运动方程。第 5 章则简要地介绍了部分常见的材料本构关系，包括非线性弹性、弹塑性、粘弹性和粘弹塑性等。这三章是非线性有限单元法的力学基础。

第 6 章介绍了线性有限单元法。非线性有限单元法是从线性有限单元法发展而来，所以本章也是非线性有限单元法的基础。第 7 章包括作者近几年的最新研究成果：退化单元和虚拟层合单元，并介绍了它们在建筑工程和机械工程中的应用。

第 8、9 章分别介绍了大变形问题、小变形材料非线性问题的有限单元法，这两章基于第 3、4、5 章非线性理论和第 6 章线性有限单元法的基础之上，是将理论转化为应用的关键。

第 10 章介绍了有限元软件设计的基本思想，这些思想大多是前人花费了巨大代价才得出的经验总结。掌握软件工程的基本思想，遵循其基本要求和规范，在编制软件时可以少走弯路，获得意想不到的效果。第 11 章比较系统地介绍非线性有限元分析软件 VFEAP。这一章将为有限元软件再开发提供一个很好的平台。第 12 章介绍了一些比较典型的工程实例。

必须指出的是，VFEAP 是在 FEAP 的基础上发展而来。FEAP 最早是美国加利福尼亚大学 R.L.Taylor 教授领导开发的非线性有限元程序，主要适用于教学和研究。1986 年 R.L.Taylor 教授和 J.C.Simo 教授受联合国教科文组织的派遣，来浙江大学讲学，并赠送了一套 FEAP 源程序。在此基础上，我们进一步研制开发了 ZD－FEAP 和 VFEAP。在此，我们对 FEAP 程序的赠送者 R.L.Taylor 教授和 J.C.Simo 教授以及在开发过程中作出过贡献的教师和研究生表示衷心感谢。

本书可作为土木、机械、力学等学科的研究生和高年级本科生的教材，也可供工程技术人员学习非线性有限元法和程序时参考。

本书第 1、第 7 章由徐兴教授编写，第 13 章由蔡金标博士编写，其余各章均由凌道盛博士编写。全书由徐兴教授统稿。

由于作者水平有限，定有不妥和错误之处，恳请专家和读者批评指正。

徐　兴

2004 年 8 月

目 录

第 1 章 绪论.....1

1.1 有限元发展历史、现状及应用前景.....1

1.2 固体力学非线性问题的分类.....2

参考文献.....4

第 2 章 有限元分析中的数值方法.....6

2.1 线性代数方程组的解法.....6

2.2 非线性方程组的解法.....8

2.3 广义特征值问题的解法.....19

2.4 动力方程的解法.....24

2.5 数值积分.....28

参考文献.....35

第 3 章 有限变形分析.....36

3.1 物体运动和变形的物质描述和空间描述.....36

3.2 变形梯度、体元、面元.....37

3.3 格林应变、阿耳曼西应变和柯西应变.....40

3.4 变形率.....45

参考文献.....48

第 4 章 应力理论和动力学基本方程.....49

4.1 Euler 应力张量.....49

4.2 Lagrange 应力张量和 Kirchhoff 应力张量.....51

4.3 微分形式的运动方程.....53
4.4 积分形式的运动方程——虚功方程.....56
4.5 应力率.....58
参考文献.....60

第 5 章 本构关系理论.....61
5.1 本构方程的基本原理.....61
5.2 弹性介质的本构关系.....62
5.3 应力空间表述的小变形弹塑性本构关系.....64
5.4 应变空间表述的小变形弹塑性本构关系.....71
5.5 几种常见的弹塑性模型.....75
5.6 粘弹性介质的本构关系.....82
5.7 Perzyna（波任纳）弹粘塑性材料.....85
参考文献.....88

第 6 章 线性有限单元法.....89
6.1 线弹性力学问题的基本方程.....89
6.2 线弹性有限单元法概述.....92
6.3 收敛准则.....98
6.4 等参数单元.....99
6.5 质量阵、阻尼阵和温度应力.....115
6.6 节点应力.....117
6.7 线性有限单元法的实施步骤与注意事项.....120
参考文献.....122

第 7 章 改进和退化的等参单元.....123
7.1 改进的等参单元.....123
7.2 改进的等参元在自由面渗流分析中的应用.....126
7.3 退化的等参单元.....128
7.4 改进的实体退化单元——组合单元.....138
7.5 组合单元的应用.....147
参考文献.....151

第 8 章 大变形问题的有限单元法.....153
8.1 有限元法方程的建立——Lagrange 法.....153

8.2 大变形问题的增量解法——T.L.法.....157
8.3 大变形问题的增量解法——U.L.法.....160
8.4 2 节点大变形杆单元.....163
8.5 2 节点大挠度平面梁单元.....177
8.6 非线性自由振动的迭代响应法.....183
参考文献.....187

第 9 章 材料非线性问题的有限单元法.....188
9.1 非线性弹性问题的有限单元法.....188
9.2 小变形弹塑性问题的有限单元法.....195
9.3 突加荷载作用下理想弹塑性梁的非线性振动.....201
9.4 节理单元.....204
9.5 小变形弹塑性有限单元分析的基本模块.....208
参考文献.....220

第 10 章 接触问题的有限单元法.....221
10.1 接触问题有限单元法的基本概念.....221
10.2 接触问题求解的一般过程.....225
10.3 接触问题的罚函数法.....228
10.4 接触问题的间隙单元法.....234
参考文献.....239

第 11 章 有限元软件方法.....240
11.1 有限元软件发展概述.....240
11.2 有限元分析的算法结构.....242
11.3 国内外有限元软件简介.....244
11.4 有限元分析软件中若干技术问题.....245
参考文献.....249

第 12 章 非线性有限元分析程序 VFEAP.....250
12.1 VFEAP 的主要功能和结构.....250
12.2 VFEAP 的主要源文件和变量.....253
12.3 VFEAP 的宏命令和算法组织.....259
12.4 主要输入输出文件及格式.....264
12.5 宏命令和单元库的扩展.....268

12.6 运行及输入输出样本.....271
参考文献 281

第 13 章 有限单元法在桥梁工程中的应用.....282
13.1 嘉王公路北郊河大桥的有限元分析.....282
13.2 宁波甬江大桥的有限元分析.....285
13.3 香港青马大桥的振动分析.....290
13.4 小结.....309
参考文献 309

第 14 章 有限单元法在土坡稳定分析中的应用.....310
14.1 边坡及其稳定性的基本概念.....310
14.2 土坡稳定分析的条分法简介.....313
14.3 利用有限单元法求解滑动面应力的土坡稳定分析.....315
14.4 土坡稳定分析的有限元强度折减法.....325
参考文献 330

附 录 VFEAP 主要源文件文本.....332
附 1 VFEAP.f90 文本.....332
附 2 Input.f90 文本.....337
附 3 Execute.f90 文本.....358
附 4 ElmLib.f90 文本.....380

第1章

绪 论

1.1 有限元发展历史、现状及应用前景

在实际工程中，对许多力学问题或场问题，人们已经得到了它们应当遵循的基本方程（常微分方程或偏微分方程）和相应的定解条件。弹性力学告诉我们，一般的固体力学问题需要满足几何关系、物理关系和平衡方程（或运动方程）。能用解析的方法求出精确解的只是方程性质比较简单、几何边界相当规则的少数问题。大多数的实际工程问题，或由于几何边界比较复杂，或由于问题的某个或某些方面表现为非线性，很难得到解析解。对于这类问题，工程和研究人员经常采用两种方法来解决。一种方法就是引入简化假定,将相对复杂的问题简化为可以通过解析方法求解的简单问题。这种方法的局限性是显而易见的，因为过多的简化可能导致不精确甚至错误的结果。另一种方法就是借助计算机利用数值方法求解。目前，数值解法已成为解决工程问题的最重要方法之一。

在所有数值解法中，有限差分法是人们较早普遍采用的数值方法。借助于有限差分技术，人们能够比较容易地得到一些复杂问题的解。目前，这一方法仍然被广大科研和工程人员采用。但是，当遇到复杂的几何形状和边界条件时，有限差分法解的精度受到限制，甚至发生求解困难。

随着电子计算机的飞速发展和广泛应用，一种新的功能强大、适用范围广泛的数值方法——有限单元法出现了。“有限单元法”（Finite Element Method，缩写为 FEM）这一名称最早是由 Clough[1] 在 1960 年的一篇关于平面弹性问题的论文中提出的。其实，有限单元法最初是在 20 世纪 50 年代作为处理固体力学问题的一种方法出现的，它是结构分析矩阵方法的一个分支。据文献记载，早在 1943 年 Courant[2] 就应用了“单元”，他在求解 St.Vnant 扭转问题时，将杆的横截面剖分为三角形“单元”，假设翘曲函数在三角形单元中呈线性分布。在 Courant 之后 10 多年，这一方法逐渐流行起来。

Besseling[25]于 1963 年将有限单元法和传统的里兹（Ritz）法比较后指出，有限单元法是里兹法的另一种形式，其试函数就是分片插值函数，在单元上解析，在整个域上仅满足连续的条件。试函数的这一改进，使有限单元法比普通的里兹法更灵活，适应性更强。目前，有限单元法在工程界获得了广泛的应用，已成为杰出的工程分析工具。

我国学者对有限单元法的创建和发展也有不少贡献。著名学者冯康在 1965 年提出的“基于变分原理的差分格式”[3]，就是有限单元法。卞学璜[4]于 1971 年指出，对某些边值问题，

有限单元法和有限差分法得出的方程组是一致的。

初期的有限单元法是建立在虚功原理或最小势能原理基础上的。在1960—1970年这十年中，发展了以各种不同变分原理为基础的有限单元法，并在许多方面对有限单元法进行了研究：

（1） 有限单元法与加权残数法的关系[5-8]；

（2） 发展了弯曲元、曲元和等参元[9-11]；

（3） 作为偏微分方程的数值解法，广泛应用于结构的非线性问题和动力学问题、固体力学问题、流体力学问题、热力学问题以及其他方法难以处理的工程问题[12]；

（4） 建立在函数分析概念上的数学基础[13, 14]。

自1967年以来，出版了许多有限单元法的专著[12, 15-18]，发行了不少有限单元法的杂志[19-24]，关于有限元理论和应用的论文也层出不穷，展现出数值方法前所未有的生命力。板壳结构有限元分析的研究尤其活跃，这是由于按板壳理论直接构造协调的简单有效的板壳单元碰到巨大的困难，研究人员提出了多种解决办法，获得一定的成功，譬如龙驭球等提出的拟协调单元等。1993年以来，作者及其研究生，从三维实体等参单元出发，直接引入相应的基本假定，构造了中厚板壳单元、Kirchhoff薄板单元、空间梁单元、平面梁单元等一系列退化单元。另外，为适应层合板壳和箱梁等复杂结构的计算，发展了层合板壳单元、虚拟层合板壳单元和组合单元[26-30]，突破了不同材料应划分为不同单元的要求。这一改进，使得复杂结构的整体空间分析、动力分析、非线性分析、形状优化等复杂问题变得简单有效。

现在有限单元法已广泛应用于航空航天、船舶建造、核能电站、地下建筑等结构工程，在潮汐运动、热传导、化学反应中物质的传递和扩散以及流体和结构的相互作用等领域，也有广泛的研究和应用。随之而来，有限单元法的专用和通用软件也大量出现，比较著名的通用软件有NASTRAN、ASKA、MARC、ANSYS、TITUS、ADINA等。

应当说有限单元法已经相当成熟，因此不大可能再有戏剧性的突破了。但是在进一步扩大应用范围、提高和完善有限单元法的使用技巧等方面还有许多工作要做。例如断裂、损伤等材料破坏问题，各种非线性问题、流体问题和大规模复杂结构的求解效率、误差界限和收敛速度的估计，网格剖分的自动化，自适应有限单元，以及有限单元和优化相结合的计算机辅助设计等等。

1.2 固体力学非线性问题的分类

从本质上讲，所有固体力学问题都是非线性的，很少有解析解，线性弹性力学问题只是实际问题的一种简化假定。在有限元分析中，线性化假设通常包含以下内容：第一，节点位移为微小量；第二，材料是线性弹性的；第三，物体运动或变形过程中，边界条件的性质保持不变。上述三个假设中的任何一个不满足，都属于非线性问题。

有鉴于此，我们通常把固体力学非线性问题分成三大类，即材料非线性、几何非线性和边界条件非线性。

一、材料非线性

材料非线性是指材料的物理定律是非线性的，简单地说，就是材料的应力应变关系是非线性的。通常，位移分量仍假设为无限小量，即应变位移间满足线性关系。材料非线性问题一般分成两大类，第一类是非线性弹性问题，橡皮、塑料、岩石等材料的应力应变关系在很小的应

变下就表现出明显的非线性性质。非线性弹性问题的一个重要特点是所有变形卸载后都是可以恢复的。另一类是具有不可逆的塑性变形的材料非线性问题，如弹塑性问题、粘弹塑性问题等，岩土工程中的软粘土就属于这类材料。

二、几何非线性

几何非线性问题一般指的是大位移问题，或者结构内部的应变较大，或者结构经历了较大的刚体位移（如平动或转动）但应变分量仍假设为微小，此时假定应力-应变关系仍然是线性的。大多数的大位移问题都属于后者，只有在材料出现塑性变形时，以及类似橡皮这样的材料才会遇到大的应变。一般的，几何非线性问题的应变和位移间不再满足线性关系。

几何非线性问题的另一个表现是，平衡方程必须相对于预先未知的变形后的几何位置给出。实际上，所有问题的平衡都是在变形后的位置上达到的，因此必须用已变形的位置写出它的平衡方程。在弹性力学中，由于假设位移很小，且假定问题的基本特征不因变形而改变，因此不需要严格区分平衡方程是基于变形前还是变形后的位置写出的。必须理解，位移的大小并不是区分几何线性和非线性的惟一标准。如图 1 示，外荷载不大时，图 1.1（a）所示结构近似为线性结构是合理的，但图 1.1（b）所示结构必须由变形后的几何位置写出。压杆失稳后的变形研究、板壳大挠度问题等均属于几何非线性问题。

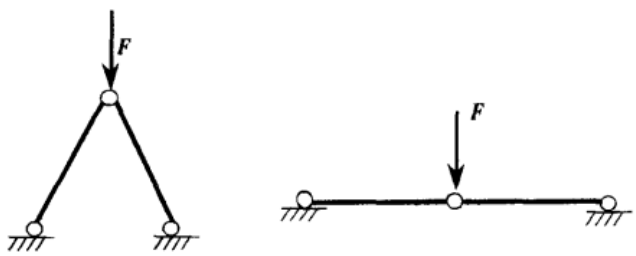


图1.1　几何非线性示意图

三、边界非线性

边界非线性问题的非线性效应是由于边界条件随物体的运动发生变化所引起的。其中最典型的例子是接触问题和随动荷载问题。实际上，几乎所有力学问题都或多或少地存在边界非线性，特别是在支座连接处，只是大部分情况下，由于这种非线性对结构的内力和变形影响很小而被忽略了。而机械加工过程中的边界非线性效应往往是不可忽略的。

关于非线性问题的分类，有两点是必须注意的：第一，不可能存在一个明确、不变的定量标准可以作为材料、几何或边界非线性的判断依据。分析人员必须依据实际的工程问题进行判断。仍以图 1.1(a)和(b)所示结构为例，两根完全相同的杆件构成不同的结构，在相同的杆件应变条件下，两者表现出来的非线性性质可能完全不同。第二，实际的非线性问题很难简单地归纳为某一类非线性，一般的情况是，物体的位移和应变都不是无限小量，本构关系也是非线性的，同时边界条件也可能随变形的发展而改变，这就大大地增加了问题分析的困难。为简化分析，抓住主要的非线性影响因素，将一些对物体运动和变形影响不大的非线性因素线性简化是必需的。

参考文献

[1]R.W.Clough.The finite element method in plane stress analysis.Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh, Pa., 1960

[2]R.Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Am. Math. Soc.*, 1943, 49(2)

[3]冯康.基于变分原理的差分格式.应用数学与计算数学，1965, 2(4)

[4]T.H.H.Pian. Variational formulation of numerical methods in solid continua, Computer-Aided Engineering, G.M.L.Gladwell,ed, University of Waterloo Press, Waterloo, Canada, 1971

[5]O.C.Zienkiewicz and G.S.Holister. Stress analysis. Wiley, New York, 1965

[6]R.E.Greene, R.E.Jones, R.W.Mclay and D.R.Strome. Generalized variational principles in the finite element method. *AIAA J.*, 1969, 7(7)

[7]B.A.Finlayson. Weighted residual methods and their relation to finite element methods in flow problems. *Finite Element in Fluids*, 1975, 2

[8]E.R.de Arantes, E. Oliverra. Theoretical foundations of the finite element method. *International Journal of Solids and Structures*, 1968, 4

[9]C.A.Felippa. Refined finite element analysis of linear and non-linear two-dimensional structures, Report US SESM 66-22, Department of Civil Engineering, University of California, Berkeley, October 1966

[10]J.G.Ergatoudis, B.M.Irons and O.C.Zienkiewicz. Three-dimensional analysis of arch dams and their foundations, Symposium on Arch Dams, Institute of Civil Engineering, London, 1968

[11]B.M.Irons and O.C. Zienkiewicz. The iso-parametric finite element system-A new concept in finite element analysis, Proceedings Conference on Recent Advances in Stress Analysis, Royal Aeronautical Society, London, 1968

[12]O.C. Zienkiewicz, The finite element method (third edtion), McGraw-Hill, New York, 1977

[13]A.K.Aziz. The mathematical foundations of the finite element method with applications to partial differential equations. New York:Academic press, 1972

[14]J.R.Whiteman. The mathematics of finite elements and applications. London: Academic press, 1973

[15]J.T.Oden. Finite elements of non-linear Continua. New York: McGraw-Hill, 1972

[16]R.D.Cook, Concepts and applications of finite element analysis, Wiley, 1974

[17]K.J.Bathe and E.L.Wilson. Numerical methods in finite element analysis. Pratice-Hall, 1976

[18]K.H.Huebuer, The Finite element method for engineers, Wiley, 1975

[19]International Journal for Numerical Methods in Engineering (eds O.C.Zienkiewicz and R.H. Gallagher), Wiley

[20]International Journal of Computers and Structures (ed. H.liebowitz), Pergamon Press

[21]Computer Methods in Applied Mechanics and Engineering (ed.J.H.Argyris), North Holland

[22]International Journal of Computers and Fluids (ed. C.Taylor), Pergamon Press

[23]International Journal of Numerical Methods in Geotechnics (ed. C.S.Desai), Wily

[24]Finite Element News, Robinson and Associates, Dorset, England

[25]J.F.Bosseling. The complete snalogy between the martin equations and the continuous fluid equations of structural analysis, International Symposium on Analogue and Digital Techniques Applied to Aeronautics. Liege, Belgium, 1963

[26]X.Xu and R.F.Cai.A new plate-shell element of 16 nodes 40 degrees of freedom by relative

displacement method. *Communications in Numerical Methods in Engineering*, 1993, 9
[27]凌道盛等.虚拟层合板单元及其在桥梁工程中的应用.土木工程学报，1998，(3)
[28]徐兴，凌道盛.实体退化单元系列.固体力学学报，2001，(计算力学专辑)
[29]李芳.结构形状和拓扑优化的虚拟层合单元法.浙江大学博士学位论文，2002
[30]蔡金标.组合单元及其在桥梁工程中的应用.浙江大学博士学位论文，2002

第2章

有限元分析中的数值方法

作为一种工程中广泛应用的数值方法，有限单元法涉及大量的基本数学问题，掌握这些问题的求解方法，有利于加深对有限元法分析过程的理解，是有效应用和开发有限元程序的基础。本章对位移法有限元分析过程中经常涉及的数值方法作一简单归纳，以供读者参考。

对于以位移为基本未知量的有限元法，力学模型经有限元离散、单元分析、系统组集和引入边界条件后，一般可以得到以下形式的运动方程，

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \tag{2.1}$$

式中，$\boldsymbol{x}$ 和 $\boldsymbol{f}$ 为 N 阶的矢量；$\boldsymbol{x}$ 是节点位移矢量，为待求量；$\boldsymbol{f}$ 为等效节点力矢量，反映了物体力的边界条件，一般为已知量。顶标一点和两点分别代表对时间的一阶和二阶导数。$\boldsymbol{M}$、$\boldsymbol{C}$ 和 $\boldsymbol{K}$ 为 $N\times N$ 的方阵，分别代表离散系统的质量阵、阻尼阵和刚度阵，适当引入位移边界条件后的系统刚度阵 $\boldsymbol{K}$ 一般为正定矩阵。注意到系统的质量阵、阻尼阵和刚度阵一般为对称且稀疏矩阵，在计算机中多采用二维等带宽或一维变带宽存贮。对于简单结构（如土木工程中围护桩的内力分析），除对角元附近的少数元素外，其他元素全部为零，此时可以采用等带宽存贮，以提高存贮和计算效率。一维变带宽存贮具有存贮紧凑、需要计算机的物理内存小的优点，但由于寻找元素较前者复杂，故不易编程且存取元素耗时较多。

对于静力学问题，方程组（2.1）简化为

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \tag{2.2}$$

对于动力学问题，加上初始条件，

$$\boldsymbol{x}\big|_{t=0} = \boldsymbol{x}_0 \qquad \dot{\boldsymbol{x}}\big|_{t=0} = \dot{\boldsymbol{x}}_0 \tag{2.3}$$

方程(2.1)便为适定，可以求得惟一解。采用数值方法对(2.1)式关于时间离散后也可得到类似（2.2）的方程组。因此，求解方程组（2.2）是有限元分析的重要步骤之一。

随着分析模型的不断复杂化和对精度要求的不断提高，代数方程组的规模越来越大，解代数方程组的时间在整个解题时间中占很大的比例，动力学问题和非线性问题尤其如此。因此，时间离散和方程组求解的方法对计算精度和速度有很大的影响，有时甚至直接关系到计算的成败。一些经典著作[1][2]均对有限元法中的数值方法进行过细致的介绍。

2.1 线性代数方程组的解法

若式（2.2）中 $\boldsymbol{K}$ 和 $\boldsymbol{f}$ 都为常数矩阵和矢量，则方程组为线性代数方程组，它的求解是非线性方程组求解的基础。线性代数、数值分析和许多有限元教材对线性代数方程组的求解都

有详细的介绍，本节只简单介绍部分方法的基本原理。

目前常用的线性方程组求解方法可分成两类：直接解法和迭代解法。直接解法以高斯消去法为基础，求解效率高，但计算过程中的舍入误差会随着方程组阶数的提高而不断累加，因此一般适用于阶数不是很高（例如不超过 10000 阶）的方程组[3]。常用的直接解法有三角分解法、分块解法和波前法等。当方程组阶次过高时，可以考虑采用迭代方法，其中常见的迭代法有高斯-赛德尔(Gauss-Seidel)迭代法、超松弛迭代法(SOR 法)等。

2.1.1 直接解法

高斯消去法是所有直接解法的基础。对于 N 阶线性方程组，系数矩阵经 $N-1$ 次消元(行变换)后化为一个上三角阵，回代计算后可得到方程组的解。与高斯消去法不同，高斯-约当消去法则是直接把系数矩阵消成一个单位矩阵，因此在形式上没有回代的过程。但实际上，高斯-约当消去法的反消过程就相当于高斯消去法的回代过程。

三角分解法则利用了三角分解定理和有限元分析中系数矩阵的对称正定性质，将系数矩阵分解为如下形式，

$$\boldsymbol{K}=\boldsymbol{L}\,\boldsymbol{D}\,\boldsymbol{L}^{T} \tag{2.4}$$

式中，$\boldsymbol{L}$ 为对角元素为 1 的下三角矩阵，$\boldsymbol{D}$ 为对角矩阵，上标 T 表示矩阵转置。可以证明这样的分解存在且惟一。分解后，分别向前、向后回代即可求得方程组的解。

与高斯消去法相比，三角分解的过程实际上相当于消去过程，两者的运算量基本相同。如果考虑系数矩阵一维变带宽存贮时取数和送数所消耗的时间，三角分解法较高斯消去法要节省时间。

为达到一定的计算精度，离散模型划分的单元一般很多，相应的节点和节点自由度很多，方程组的阶次很高。即使采用一维变带宽存贮，系数矩阵有时也不能全部进入内存。分块解法和波前法是解决计算机内存不足的有效方法。

分块解法和波前法的基本思想都是基于对高斯消去法的再分析上，由先组集后消元发展到组集和消元交替进行[3]。分块解法采用自由度组集完一批消去一批的方法，而波前法采用组集完一个自由度消去一个的方法。与分块解法比较，波前法需要的内存更小，但程序编制复杂，内外存交换频繁。基于组集和消元交替进行思想派生出的解法不少，不同的解法各有特点，但基本思想相同。

2.1.2 迭代解法

雅可比(Jacobi)迭代法和高斯-赛德尔(Gauss-Seidel)迭代法是求解线性方程组的常用迭代法，两种方法求解思路是相似的。

将(2.2)式改写成如下形式，

$$x_i=\frac{1}{K_{ii}}\left(f_i-\sum_{\substack{j=1\\ j\neq i}}^{N}K_{ij}x_j\right)\quad (i=1,2,\cdots,N) \tag{2.5}$$

式中，x_i，f_i 分别为矢量 $\boldsymbol{x}$ 和 $\boldsymbol{f}$ 的第 i 个分量，K_{ij} 为矩阵 $\boldsymbol{K}$ 的第 i 行第 j 列对应的元素，下同。本章中，同一项中相同指标不满足爱因斯坦求和约定。

假设 $\boldsymbol{x}$ 的一个初值 $\boldsymbol{x}^{(0)}$，由(2.5)式可以构造如下迭代格式，

$$x_i^{(n+1)}=\frac{1}{K_{ii}}(f_i-\sum_{\substack{j=1\\ j\neq i}}^{N}K_{ij}x_j^{(n)}) \quad (i=1,2,\cdots,N,\ n=0,1,\cdots) \tag{2.6}$$

这就是雅可比迭代法。要注意的是，由于不能保证总是收敛的，因此不常采用该方法求解有限元线性方程组。

在迭代格式(2.6)中，由于计算是顺序进行的，在计算某一个 $x_k^{(n+1)}$ 时，其前面的分量 $x_1^{(n+1)}$，$x_2^{(n+1)}$，…，$x_{k-1}^{(n+1)}$ 已经求得，如果迭代过程收敛，它们应该比 $x_1^{(n)},x_2^{(n)},\ldots,x_{k-1}^{(n)}$ 更接近准确值，用前者代替应该可以起到加速收敛的作用。于是有

$$x_i^{(n+1)}=\frac{1}{K_{ii}}(f_i-\sum_{j=1}^{i-1}K_{ij}x_j^{(n+1)}-\sum_{j=i+1}^{N}K_{ij}x_j^{(n)}) \quad (i=1,2,\cdots,N,\ n=0,1,\cdots) \tag{2.7}$$

这就是著名的高斯-赛德尔迭代法。

理论分析表明，当线性方程组的系数矩阵为对称正定矩阵时，高斯-赛德尔迭代法一定收敛。因此，它成为有限元法中求解线性代数方程组的常用迭代方法。

为提高迭代法的收敛速度，将(2.7)式改写成如下形式，

$$x_i^{(n+1)}=x_i^{(n)}+\frac{1}{K_{ii}}(f_i-\sum_{j=1}^{i-1}K_{ij}x_j^{(n+1)}-\sum_{j=i}^{N}K_{ij}x_j^{(n)}) \quad (i=1,2,\cdots,N,\ n=0,1,\cdots) \tag{2.8}$$

上式右端第二项代表两次迭代间的增量，引入系数 ω 得

$$x_i^{(n+1)}=x_i^{(n)}+\frac{\omega}{K_{ii}}(f_i-\sum_{j=1}^{i-1}K_{ij}x_j^{(n+1)}-\sum_{j=i}^{N}K_{ij}x_j^{(n)}) \quad (i=1,2,\cdots,N,\ n=0,1,\cdots) \tag{2.9}$$

式(2.9)就是超松弛迭代法(简称 SOR 法)的计算公式，系数 ω 叫做松弛因子。可以证明，为保证迭代过程的收敛性，必须要求 $0<\omega<2$。为达到加速收敛的目的，常取 $\omega>1$。显然 $\omega=1$ 时，式(2.9)就退化为高斯-赛德尔迭代。

2.2 非线性方程组的解法

若式（2.2）中 $\boldsymbol{K}$ 或 $\boldsymbol{f}$ 为 $\boldsymbol{x}$ 和（或）$\boldsymbol{x}$ 关于时间导数的函数，则方程组为非线性方程组。非线性方程组一般不可以直接求解，通常以一系列线性代数方程组的解去逼近，因此求解要复杂、耗时得多。不同的线性逼近方法对应不同的非线性方程组的求解方法。目前常用的非线性方程组求解方法有：直接迭代法、牛顿法(也称为 Newton-Raphson 法或切线刚度法，简称为 N-R 法)、修正的牛顿法(简称为 mN-R 法)、拟牛顿法、载荷增量法和弧长法等。

直接迭代法是一种最简单、最直观的方法。由于存在收敛速度慢、迭代过程不稳定、严重依赖于初值的选取等缺点，直接迭代法在实际应用中很少采用。牛顿法求解非线性方程组具有收敛速度快的优点，修正的牛顿法和拟牛顿法则提高了计算效率。弧长法克服了牛顿法不能越过极值点的缺点，比较适合于分析结构软化问题。

2.2.1　直接迭代法

设 $\boldsymbol{x}$ 的某个初始近似值为 $\boldsymbol{x}^{(0)}$，相应的系数矩阵 $\boldsymbol{K}^{(0)}$ 可以按下式得到，

$$\boldsymbol{K}^{(0)}=\boldsymbol{K}(\boldsymbol{x}^{(0)})$$

由（2.2）式可以得到一个改进的近似值，

$$\boldsymbol{x}^{(1)}=(\boldsymbol{K}^{(0)})^{-1}\boldsymbol{f}$$

重复这样的过程，从第 n 次近似值到第 n+1 次近似值的求解公式为

$$\begin{cases}\boldsymbol{K}^{(n)}=\boldsymbol{K}(\boldsymbol{x}^{(n)})\\ \boldsymbol{x}^{(n+1)}=(\boldsymbol{K}^{(n)})^{-1}\boldsymbol{f}\end{cases}\tag{2.10}$$

不断重复上述迭代过程，直到相邻两次迭代的计算值的“偏差”，

$$\Delta\boldsymbol{x}^{(n)}=\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^{(n)}$$

充分小为止。迭代过程的收敛性可以采用以下三类准则进行判断。

一、位移收敛准则

位移收敛准则利用近似解偏差的某种范数作为迭代收敛的判断标准。理论上说，近似解偏差的任意一种范数都可以作为收敛判断的依据，但在实际的有限元分析中，由于具有明确的物理意义便于控制，“偏差”的“∞”范数和“2”范数被广泛采用。即

$$\left\|\Delta\boldsymbol{x}^{(n)}\right\|_{\infty}=\max_{1\leqslant i\leqslant N}\left|\Delta x_i^{(n)}\right|\leqslant\alpha_1\left\|\boldsymbol{x}^{(n+1)}\right\|_{\infty}\tag{2.11}$$

$$\left\|\Delta\boldsymbol{x}^{(n)}\right\|_{2}=\left[(\Delta\boldsymbol{x}^{(n)})^{T}\cdot\Delta\boldsymbol{x}^{(n)}\right]^{1/2}\leqslant\alpha_2\left\|\boldsymbol{x}^{(n+1)}\right\|_{2}\tag{2.12}$$

其中，α_1 和 α_2 为事先定义的大于零的小数，代表规定的容许误差。应用中要注意两点，第一，相同容许误差的条件下，式（2.11）和（2.12）对应的收敛标准是不同的，需要的迭代次数可能相差很多；第二，弹塑性有限元分析中应小心使用位移收敛标准，容许误差不宜过小，否则可能会出现迭代不易收敛的现象。

二、平衡收敛准则

对于每个迭代步的近似值 $\boldsymbol{x}^{(n)}$，可以求得

$$\boldsymbol{\Psi}^{(n)}=\boldsymbol{K}(\boldsymbol{x}^{(n)})-\boldsymbol{f}\tag{2.13}$$

一般地，$\boldsymbol{\Psi}^{(n)}\neq\boldsymbol{0}$，它代表了对平衡点偏离的一种度量，称为失衡力。平衡收敛准则就是利用这一失衡力的某种范数作为判断标准。

$$\left\|\boldsymbol{\Psi}^{(n)}\right\|_{\infty}\leqslant\beta_1\left\|\boldsymbol{f}\right\|_{\infty}\tag{2.14}$$

$$\left\|\boldsymbol{\Psi}^{(n)}\right\|_{2}\leqslant\beta_2\left\|\boldsymbol{f}\right\|_{2}\tag{2.15}$$

β_1 和 β_2 为事先指定的大于零的小数，代表失衡力的容许误差。

三、能量收敛准则

一些文献也建议采用如下形式的能量收敛准则，

$$(\boldsymbol{\Psi}^{(n)})^{T}\Delta\boldsymbol{x}^{(n)}\leqslant\gamma\boldsymbol{f}^{T}\Delta\boldsymbol{x}^{(n)}\tag{2.16}$$

式中，γ 为事先指定的一个大于零的小量。

对于单变量的非线性方程，直接迭代法的计算过程如图 2.1 示。图中，$P(x)=K(x)\cdot x$。

直接迭代法的优点是方法简单、易于编程，缺点是迭代过程不一定收敛，其收敛与否还与初始近似值的选取有关。即使收敛，收敛速度一般较慢。为避免迭代不收敛而陷入死循环，启动迭代分析之前应设定一个最大的迭代步数。

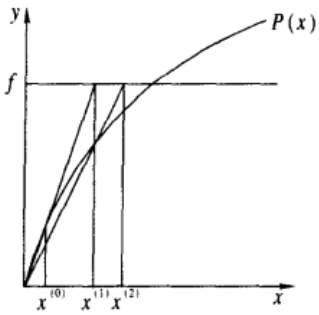


图 2.1 单变量非线性方程直接迭代法

2.2.2 牛顿法和修正的牛顿法

牛顿法又称牛顿-拉斐逊（Newton-Raphson）法，力学上称之为切线刚度法，是最著名的非线性方程组求解方法之一。

为描述方便起见，（2.2）式可以改写为如下形式，

$$\boldsymbol{\Psi}(\boldsymbol{x})=\boldsymbol{K}(\boldsymbol{x})\boldsymbol{x}-\boldsymbol{f}=\boldsymbol{0} \tag{2.17}$$

设 $\boldsymbol{\Psi}(\boldsymbol{x})$ 为具有一阶导数的连续函数，初始近似值为 $\boldsymbol{x}^{(0)}$，第 n 次迭代的近似值为 $\boldsymbol{x}^{(n)}$。把 $\boldsymbol{\Psi}(\boldsymbol{x})$ 在 $\boldsymbol{x}^{(n)}$ 处泰勒展开，保留线性项，忽略高阶项得

$$\boldsymbol{\Psi}(\boldsymbol{x}^{(n)})+\boldsymbol{K}_T^{(n)}(\boldsymbol{x}-\boldsymbol{x}^{(n)})\approx\boldsymbol{0}$$

求解上式得到 $\boldsymbol{x}$ 的一个新的近似值 $\boldsymbol{x}^{(n+1)}$，

$$\boldsymbol{x}^{(n+1)}=\boldsymbol{x}^{(n)}-(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{\Psi}(\boldsymbol{x}^{(n)}) \tag{2.18}$$

式中，$\boldsymbol{K}_T^{(n)}=\left.\dfrac{\partial\boldsymbol{\Psi}}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}}$，在位移法有限元分析中代表结构的切线刚度矩阵，它由相应的单元切线刚度矩阵组集而成。通过与上一迭代步得到的结果进行相对误差比较，可以控制收敛过程。单自由度问题的迭代过程如图 2.2 所示。

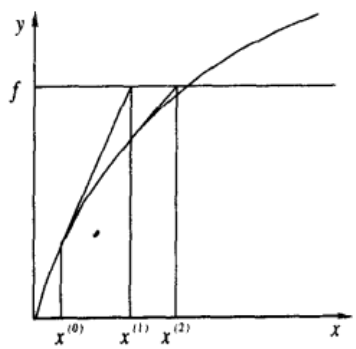


图2.2 Newton法迭代示意图

Newton 法的求解步骤为：

(1) 设初始值 $\boldsymbol{x}^{(0)}$，令 n=0；

(2) 计算切线刚度矩阵，

$$\boldsymbol{K}_T^{(n)}=\left.\frac{\partial\boldsymbol{\Psi}}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}}$$

(3) 计算不平衡量，

$$\boldsymbol{\Psi}^{(n)}=\boldsymbol{\Psi}(\boldsymbol{x}^{(n)})=\boldsymbol{K}(\boldsymbol{x}^{(n)})\boldsymbol{x}^{(n)}-\boldsymbol{f}$$

(4) 解方程组

$$\boldsymbol{K}_T^{(n)}\Delta\boldsymbol{x}^{(n)}=-\boldsymbol{\Psi}^{(n)}$$

得

$$\Delta \boldsymbol{x}^{(n)} = -(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{\Psi}^{(n)}$$

(5) 计算 n+1 次迭代近似值，

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \Delta \boldsymbol{x}^{(n)}$$

(6) 判断是否收敛，如果收敛，迭代结束，否则令 n=n+1，转求解步骤(2)。

可以证明，牛顿法求解方程单根时的收敛阶不小于 2，一般情况具有很快的收敛速度。但在有些非线性问题（如理想弹塑问题、结构软化问题）中，由于切线刚度矩阵可能奇异或是病态，矩阵求逆有一定困难，此时可以通过引入阻尼因子的方法进行修正，以减弱切线刚度阵的病态性质。

虽然牛顿法具有收敛速度快的优点，但它要求每次迭代都重新计算切线刚度矩阵，并分解回代，计算量太大，求解效率低。除一些规模比较小的问题之外，牛顿法在有限元分析中并不多用。

为兼顾计算效率和收敛速度，修正的 Newton 法认为，迭代过程中没有必要每次都重新计算切线刚度阵，可以选择一个恰当的正整数 M，规定计算过程中每经过 M 次迭代修正一次刚度。显然，当 M=1 时，修正的牛顿法即为牛顿法；当 $M=\infty$ 时，相当于整个迭代过程中一直采用初始的切线刚度矩阵，只需分解一次。这种迭代方法称为初刚度法。经过适当的变换，还可以由此得到常用的初应力法和初应变法。相关的内容将在以后的材料非线性有限元分析中介绍。

修正 Newton 法的计算步骤为：

(1) 设定 M 和初始值 $\boldsymbol{x}^{(0)}$，令 n=0；

(2) 计算切线刚度矩阵，

$$\boldsymbol{K}_T^{(n)} = \left.\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}}$$

(3) 对 j 从 1 到 M 循环，完成

(a) 计算不平衡量，

$$\boldsymbol{\Psi}^{(n)} = \boldsymbol{\Psi}(\boldsymbol{x}^{(n)}) = \boldsymbol{K}(\boldsymbol{x}^{(n)})\boldsymbol{x}^{(n)} - \boldsymbol{f}$$

(b) 解方程组

$$\boldsymbol{K}_T^{(n)}\Delta \boldsymbol{x}^{(n)} = -\boldsymbol{\Psi}^{(n)}$$

得

$$\Delta \boldsymbol{x}^{(n)} = -(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{\Psi}^{(n)}$$

(c) 计算 n+1 次近似值，

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \Delta \boldsymbol{x}^{(n)}$$

(d) 判断是否收敛，如果收敛，迭代结束，否则令 n=n+1。

(4) 转计算步骤(2)继续迭代。

2.2.3　拟 Newton 法

Newton 法虽然有较快的收敛速度，但每次迭代要重新生成并分解总体刚度阵，而对矩阵进行分解需占用大量的计算工作量，计算效率低。为提高计算效率，拟 Newton 法采用了不同于修正 Newton 法的改进思路。在每次迭代后，拟 Newton 法既不需重新生成切线刚度阵，也

不是简单地沿用前面的切线刚度阵，而是用一种简单的方式直接对切线刚度阵的逆矩阵加以修正，从而大大减少了计算机的计算量，提高了计算效率。由此可见，拟 Newton 法的关键就在于如何选择一个简单的切线刚度阵的逆矩阵的修正方法。

假设第 n 和 $n+1$ 次迭代的近似值 $\boldsymbol{x}^{(n)}$、$\boldsymbol{x}^{(n+1)}$ 已求得，则第 $n+1$ 步的切线刚度阵 $\boldsymbol{K}_T^{(n+1)}$ 可以类似差商的性质定义如下，

$$\boldsymbol{K}_T^{(n+1)}(\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^{(n)})\approx\boldsymbol{\Psi}(\boldsymbol{x}^{(n+1)})-\boldsymbol{\Psi}(\boldsymbol{x}^{(n)}) \tag{2.19}$$

称方程（2.19）为拟 Newton 方程。$\boldsymbol{K}_T^{(n+1)}$ 可以表示为如下增量形式，

$$\boldsymbol{K}_T^{(n+1)}=\boldsymbol{K}_T^{(n)}+\Delta\boldsymbol{K}_T^{(n)} \tag{2.20}$$

设修正矩阵 $\Delta\boldsymbol{K}_T^{(n)}$ 的秩为 m（$m\geq1$）。根据线性代数的知识可知，$\Delta\boldsymbol{K}_T^{(n)}$ 可以分解为如下形式，

$$\Delta\boldsymbol{K}_T^{(n)}=\boldsymbol{U}^{(n)}(\boldsymbol{V}^{(n)})^T \tag{2.21}$$

其中，$\boldsymbol{U}^{(n)}$ 和 $\boldsymbol{V}^{(n)}$ 均为秩为 m 的 $N\times m$ 阶矩阵，N 为 $\boldsymbol{K}_T$ 的阶数。把（2.21）代入（2.20）式得

$$\boldsymbol{K}_T^{(n+1)}=\boldsymbol{K}_T^{(n)}+\boldsymbol{U}^{(n)}(\boldsymbol{V}^{(n)})^T \tag{2.22}$$

根据 Sherman-Morrison-Woodburg 公式，有

$$(\boldsymbol{K}_T^{(n+1)})^{-1}=(\boldsymbol{K}_T^{(n)})^{-1}-(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{U}^{(n)}[\boldsymbol{I}+(\boldsymbol{V}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{U}^{(n)}]^{-1}(\boldsymbol{V}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1} \tag{2.23}$$

于是，拟 Newton 公式可以表示为

$$\begin{cases}\boldsymbol{x}^{(n+1)}=\boldsymbol{x}^{(n)}+\Delta\boldsymbol{x}^{(n)}\\ \Delta\boldsymbol{x}^{(n)}=-(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{\Psi}(\boldsymbol{x}^{(n)})\\ (\boldsymbol{K}_T^{(n+1)})^{-1}(\boldsymbol{\Psi}^{(n+1)}-\boldsymbol{\Psi}^{(n)})=\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^{(n)}\\ (\boldsymbol{K}_T^{(n+1)})^{-1}=(\boldsymbol{K}_T^{(n)})^{-1}+\Delta(\boldsymbol{K}_T^{(n)})^{-1}\end{cases} \tag{2.24}$$

其中，$\Delta(\boldsymbol{K}_T^{(n)})^{-1}=-(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{U}^{(n)}[\boldsymbol{I}+(\boldsymbol{V}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\boldsymbol{U}^{(n)}]^{-1}(\boldsymbol{V}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}$，可以证明 $\Delta(\boldsymbol{K}_T^{(n)})^{-1}$ 的秩也为 m。

在实际应用中，m 常取 1 或 2，以下以 $\Delta\boldsymbol{K}_T^{(n)}$ 的秩为 2 的算法为例进一步阐述如何构造修正矩阵 $\Delta\boldsymbol{K}_T^{(n)}$ 或 $\Delta(\boldsymbol{K}_T^{(n)})^{-1}$。当 $m=2$ 时，

$$\boldsymbol{U}^{(n)}=[\boldsymbol{U}_1^{(n)},\boldsymbol{U}_2^{(n)}]\quad\boldsymbol{V}^{(n)}=[\boldsymbol{V}_1^{(n)},\boldsymbol{V}_2^{(n)}]$$

式中，$\boldsymbol{U}_1^{(n)},\boldsymbol{U}_2^{(n)},\boldsymbol{V}_1^{(n)},\boldsymbol{V}_2^{(n)}$ 为 N 维列矢量，$\boldsymbol{U}_1^{(n)}$ 和 $\boldsymbol{U}_2^{(n)}$ 互不相关，$\boldsymbol{V}_1^{(n)}$ 和 $\boldsymbol{V}_2^{(n)}$ 互不相关，将上式代入 $\Delta\boldsymbol{K}_T^{(n)}$ 可得

$$\Delta\boldsymbol{K}_T^{(n)}=\boldsymbol{U}_1^{(n)}(\boldsymbol{V}_1^{(n)})^T+\boldsymbol{U}_2^{(n)}(\boldsymbol{V}_2^{(n)})^T \tag{2.25}$$

注意到 $\Delta(\boldsymbol{K}_T^{(n)})^{-1}$ 的秩也为 m，因此我们可以类似地得到，

$$\Delta(\boldsymbol{K}_T^{(n)})^{-1}=\boldsymbol{w}_1^{(n)}(\boldsymbol{z}_1^{(n)})^T+\boldsymbol{w}_2^{(n)}(\boldsymbol{z}_2^{(n)})^T \tag{2.26}$$

式中，$\boldsymbol{w}_1^{(n)},\boldsymbol{w}_2^{(n)},\boldsymbol{z}_1^{(n)},\boldsymbol{z}_2^{(n)}$ 为 N 维列矢量，$\boldsymbol{w}_1^{(n)}$ 和 $\boldsymbol{w}_2^{(n)}$ 互不相关，$\boldsymbol{z}_1^{(n)}$ 和 $\boldsymbol{z}_2^{(n)}$ 互不相关。

把（2.25）代入拟 Newton 方程（2.19）式，得

$$\boldsymbol{U}_1^{(n)}(\boldsymbol{V}_1^{(n)})^T\Delta\boldsymbol{x}^{(n)}+\boldsymbol{U}_2^{(n)}(\boldsymbol{V}_2^{(n)})^T\Delta\boldsymbol{x}^{(n)}=\Delta\boldsymbol{\Psi}^{(n)}-\boldsymbol{K}_T^{(n)}\Delta\boldsymbol{x}^{(n)}$$

若 $(\boldsymbol{V}_1^{(n)})^T\Delta\boldsymbol{x}^{(n)}$ 和 $(\boldsymbol{V}_2^{(n)})^T\Delta\boldsymbol{x}^{(n)}$ 均不为零（标量），则可取

$$\boldsymbol{U}_1^{(n)}=\frac{\Delta\boldsymbol{\Psi}^{(n)}}{(\boldsymbol{V}_1^{(n)})^T\Delta\boldsymbol{x}^{(n)}}\qquad \boldsymbol{U}_2^{(n)}=-\frac{\boldsymbol{K}_T^{(n)}\Delta\boldsymbol{x}^{(n)}}{(\boldsymbol{V}_2^{(n)})^T\Delta\boldsymbol{x}^{(n)}}\tag{2.27}$$

把（2.26）式代入（2.24）式可得

$$\boldsymbol{w}_1^{(n)}(\boldsymbol{z}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}+\boldsymbol{w}_2^{(n)}(\boldsymbol{z}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}=\Delta\boldsymbol{x}^{(n)}-(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}$$

若$(\boldsymbol{z}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}$和$(\boldsymbol{z}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}$不等于 0（标量），则可取，

$$\boldsymbol{w}_1^{(n)}=\frac{\Delta\boldsymbol{x}^{(n)}}{(\boldsymbol{z}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\qquad \boldsymbol{w}_2^{(n)}=-\frac{(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}}{(\boldsymbol{z}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\tag{2.28}$$

于是$\Delta(\boldsymbol{K}_T^{(n)})^{-1}$可表示为

$$\Delta(\boldsymbol{K}_T^{(n)})^{-1}=\frac{\Delta\boldsymbol{x}^{(n)}(\boldsymbol{z}_1^{(n)})^T}{(\boldsymbol{z}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}-\frac{(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}(\boldsymbol{z}_2^{(n)})^T}{(\boldsymbol{z}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\tag{2.29}$$

为使上式具有更普遍的意义，作如下变换

$$\overline{\boldsymbol{z}}_1^{(n)}=\frac{\boldsymbol{z}_1^{(n)}}{(\boldsymbol{z}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\qquad \overline{\boldsymbol{z}}_2^{(n)}=\frac{\boldsymbol{z}_2^{(n)}}{(\boldsymbol{z}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\tag{2.30}$$

则显然有

$$(\overline{\boldsymbol{z}}_1^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}=(\overline{\boldsymbol{z}}_2^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}=1\tag{2.31}$$

于是（2.29）式可化为

$$\Delta(\boldsymbol{K}_T^{(n)})^{-1}=\Delta\boldsymbol{x}^{(n)}(\overline{\boldsymbol{z}}_1^{(n)})^T-(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}(\overline{\boldsymbol{z}}_2^{(n)})^T\tag{2.32}$$

任意选择满足条件(2.30)式且互不相关的$\overline{\boldsymbol{z}}_1^{(n)}$和$\overline{\boldsymbol{z}}_2^{(n)}$都可保证$\Delta\boldsymbol{K}_T^{(n)}$的秩为 2，且满足拟 Newton 方程。取方法参数$\rho^{(n)}$，按如下形式选择$\overline{\boldsymbol{z}}_1^{(n)}$和$\overline{\boldsymbol{z}}_2^{(n)}$，

当$(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}\neq 0$时，

$$(\overline{\boldsymbol{z}}_1^{(n)})^T=(1+\rho^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)})\frac{(\Delta\boldsymbol{x}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}-\rho^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\tag{2.33}$$

当$(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}\neq 0$时，

$$(\overline{\boldsymbol{z}}_2^{(n)})^T=(1-\rho^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T\Delta\boldsymbol{x}^{(n)})\frac{(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}}{(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}}+\rho^{(n)}(\Delta\boldsymbol{x}^{(n)})^T\tag{2.34}$$

把（2.33）式和（2.34）式代入（2.32）式，整理得

$$\begin{aligned}(\boldsymbol{K}_T^{(n+1)})^{-1}=(\boldsymbol{K}_T^{(n)})^{-1}&-\frac{(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}}{(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}}+\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\\&+\rho^{(n)}\left(\frac{(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}}{(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}}\times(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}\right.\\&-(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T-\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\\&\left.+\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}\right)\end{aligned}\tag{2.35}$$

我们称递推关系式（2.35）为单参数秩 2 拟 Newton 算法。选择不同的参数$\rho^{(n)}$，可以得到不同的秩 2 算式。

如果令$\rho^{(n)}=0$，则得到 DFP（Davidon-Fletcher-Powell）公式，

$$(\boldsymbol{K}_T^{(n+1)})^{-1}=(\boldsymbol{K}_T^{(n)})^{-1}+\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}-\frac{(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}}{(\Delta\boldsymbol{\Psi}^{(n)})^T(\boldsymbol{K}_T^{(n)})^{-1}\Delta\boldsymbol{\Psi}^{(n)}} \tag{2.36}$$

如果令 $\rho^{(n)}=1/(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}$，则得到 BFGS（Broyden-Fletcher-Goldfarb-Shanno）公式，

$$(\boldsymbol{K}_T^{(n+1)})^{-1}=\left(\boldsymbol{I}-\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\right)(\boldsymbol{K}_T^{(n)})^{-1}\left(\boldsymbol{I}-\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{\Psi}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}}\right)+\frac{\Delta\boldsymbol{x}^{(n)}(\Delta\boldsymbol{x}^{(n)})^T}{(\Delta\boldsymbol{x}^{(n)})^T\Delta\boldsymbol{\Psi}^{(n)}} \tag{2.37}$$

实践证明，BFGS 算法是目前最为成功的秩 2 算法之一，它有较好的数值稳定性。

2.2.4 载荷增量法

载荷增量法是不同于直接迭代法和牛顿法的又一类线性化方法。这一方法从问题的初值出发，随着外部荷载 $\boldsymbol{f}$ 按增量形式逐步增大来研究结构的运动和变形。载荷增量法可以得到整个荷载变化过程中的研究对象的运动和变形。因此特别适合与加载历史有关的力学问题（如弹塑性问题）的求解。

对于保守力系，令 $\boldsymbol{f}=\lambda\bar{\boldsymbol{f}}$，式中 λ 是一个用以描述载荷变化的参数，称为载荷参数，一般来说它与时间相关；$\bar{\boldsymbol{f}}$ 为标准载荷向量，一般只是空间函数。式(2.17)可以表示为

$$\boldsymbol{\psi}(\boldsymbol{x},\lambda)=\boldsymbol{P}(\boldsymbol{x})-\lambda\bar{\boldsymbol{f}}=\boldsymbol{K}(\boldsymbol{x})\boldsymbol{x}-\lambda\bar{\boldsymbol{f}}=\boldsymbol{0} \tag{2.38}$$

对上式求全微分，

$$\boldsymbol{K}_T\mathrm{d}\boldsymbol{x}-\mathrm{d}\lambda\bar{\boldsymbol{f}}=\boldsymbol{0}$$

即

$$\boldsymbol{K}_T\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\lambda}=\bar{\boldsymbol{f}} \tag{2.39}$$

将时间离散，各时间分点对应的荷载参数依次为 $\lambda^0,\lambda^1,\cdots,\lambda^M$。设 $\lambda=\lambda^0$ 时方程（2.2）的解 $\boldsymbol{x}^0$ 已知，

$$\boldsymbol{x}\big|_{\lambda=\lambda^0}=\boldsymbol{x}^0 \tag{2.40}$$

式(2.39)和(2.40)是一个典型的一阶微分方程组的初值问题，可以采用很多种方法求解，最简单有效的是显式欧拉法，

$$\begin{cases}\boldsymbol{x}^{m+1}=\boldsymbol{x}^m+\Delta\lambda^m(\boldsymbol{K}_T^m)^{-1}\bar{\boldsymbol{f}}\\ \boldsymbol{K}_T^m=\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\Big|_{x=x^m}\\ \Delta\lambda^m=\lambda^{m+1}-\lambda^m\end{cases}\qquad(m=0,1,2,\cdots) \tag{2.41}$$

为提高计算精度，可以采用各种形式的 Runge-Kutta 法来求解上述问题。如两点的预估-校正公式就可获得较好数值结果，但数值计算量可能会成倍增加。

图 2.3 为一维问题 Euler 算法的迭代过程。由图可以看出，Euler 算法在每一步计算中都会引起某些偏差，造成对真解的漂移，而且这种漂移可能不断地积累。为克服解的漂移现象，在实际应用中需要对 Euler 法进行修正。修正方法之一是把上一步

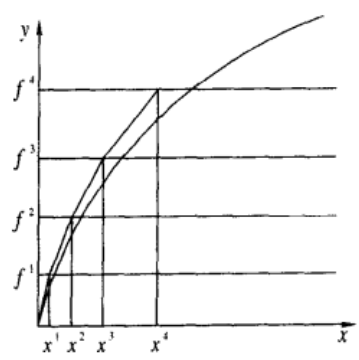


图2.3 Euler法迭代示意图

的不平衡量引入当步迭代，即将(2.41)改写为

$$\begin{cases} \boldsymbol{x}^{m+1} = \boldsymbol{x}^m + (\boldsymbol{K}_T^m)^{-1}(\lambda^{m+1}\bar{\boldsymbol{f}} - \boldsymbol{P}(\boldsymbol{x}^m)) \\ \boldsymbol{K}_T^m = \left.\dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{x=x^m} \\ \Delta\lambda^m = \lambda^{m+1} - \lambda^m \end{cases} \quad (m = 0,1,2,\cdots) \tag{2.42}$$

方法之二是把载荷增量法和牛顿法或修正的牛顿法结合在一起，即在每一个载荷步中采用牛顿法或修正的牛顿法进行迭代分析，直到当前载荷步达到收敛标准才进入下一载荷步分析。载荷增量法和修正 Newton 法结合求解非线性方程组的步骤为：

(1) 令 $m=0$；

(2) 计算切线刚度矩阵，$\boldsymbol{K}_T^m = \left.\dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{x=x^m}$，并求逆；

(3) 令 $n=0$，$\boldsymbol{x}^{m+1(0)} = \boldsymbol{x}^m$；

(4) 计算不平衡量 $\boldsymbol{\Psi}^{m+1(n)} = \boldsymbol{\Psi}(\boldsymbol{x}^{m+1(n)}) = \boldsymbol{K}(\boldsymbol{x}^{m+1(n)})\boldsymbol{x}^{m+1(n)} - \lambda^{m+1}\bar{\boldsymbol{f}}$；

(5) 解方程组 $\boldsymbol{K}_T^m \Delta\boldsymbol{x}^{m+1(n)} = -\boldsymbol{\Psi}^{m+1(n)}$，得 $\Delta\boldsymbol{x}^{m+1(n)} = -(\boldsymbol{K}_T^m)^{-1}\boldsymbol{\Psi}^{m+1(n)}$；

(6) 计算迭代步近似值，$\boldsymbol{x}^{m+1(n+1)} = \boldsymbol{x}^{m+1(n)} + \Delta\boldsymbol{x}^{m+1(n)}$；

(7) 判断是否收敛。如果收敛，令 $\boldsymbol{x}^{m+1} = \boldsymbol{x}^{m+1(n+1)}$，$m = m+1$ 转求解步骤(2)进行下一载荷步分析，直到分析完成。否则令 $n=n+1$，转求解步骤(4)进行下一迭代步分析。

与载荷增量法对应的另一种增量解法是位移增量法，由于位移增量法存在一些局限性，在工程有限元分析中并不常用，这里不再介绍。

2.2.5　弧长法

在用增量迭代法求解非线性有限元问题时，一般采用 Newton-Raphson 法，并通常假设荷载步增量或位移增量在迭代过程中保持不变，两者均存在各自的局限性。如图 2.4 示，荷载水平不变的增量迭代法是求 P 为常数的水平线与解曲线相交的交点，由这些交点来定义整个解。当有限元模型出现局部软化、平衡路径分歧或极值点时，可能会出现解（交点）不存在、不惟一或解虽存在但不属于我们意欲求解的范围。采用位移不变的增量迭代法不仅存在类似的困难，而且由于位移自由度很多，位移增量很难控制。过去二三十年来，人们一直在探索新的迭代控制方法来解决邻近极值点的求解问题，这些方法主要有：人工弹簧法、压缩平衡迭代法和弧长法，而其中又以弧长法最为有效。

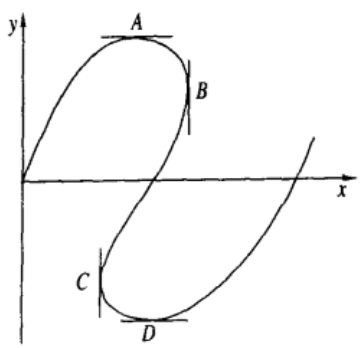


图 2.4　增量法求解策略

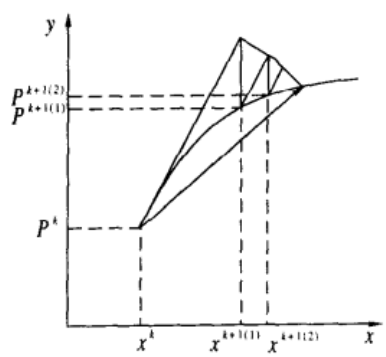


图 2.5　弧长法求解过程

弧长法最初是由 Riks 和 Wempner 提出的，继而由 Crisfield 和 Ramm 加以修正和发展。自弧长法建立以来已广泛应用于结构非线性分析，它克服了传统牛顿迭代法无法跨越结构非线性平衡路径上极值点的困难，能够在迭代求解过程中自动调节增量步长，跟踪各种复杂的非线性平衡路径全过程，例如跳跃(snap through)和跳回(snap back)现象。弧长法是目前结构非线性分析中数值计算最稳定、计算效率最高且最可靠的迭代控制方法之一。宋二祥等[6]将弧长法应用于土坡安全系数的有限元分析中。

弧长法是求取与解曲线正交的线族与解曲线相交的交点（如图 2.5 示）。由于解曲线自身未知，其正交线族实际上也是未知的。但在迭代过程中可将上次增量的收敛解或上次迭代解出的路径之切线近似地作为下次迭代解的切线。

设载荷参数为 λ^k 时位移 $\boldsymbol{x}^k$ 已知，载荷参数增量为 $\Delta\lambda^k$ 时，相应的位移增量为 $\Delta\boldsymbol{x}^k$，则

$$\boldsymbol{\psi}(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k,\lambda^k+\Delta\lambda^k)=\boldsymbol{0} \tag{2.43}$$

对于弧长法而言，(2.43)式中的 $\Delta\boldsymbol{x}^k$ 和 $\Delta\lambda^k$ 均是未知的，有 $N+1$ 个未知量，而方程数只有 N 个。求解这 $N+1$ 个未知量需要补充一个附加条件，即辅助方程，

$$f(\Delta\boldsymbol{x}^k,\ \Delta\lambda^k)=\Delta l^k \tag{2.44}$$

式中，Δl^k 为增量弧长。联立求解(2.43)和(2.44)式可求得既满足平衡条件，又满足辅助方程的 $\Delta\boldsymbol{x}^k$ 和 $\Delta\lambda^k$。采用不同的辅助方程可得到不同的弧长法，但每种方法都涉及四个部分：参数控制（即辅助方程）、弧长增量控制、初始荷载增量控制参数符号确定和荷载增量控制参数求解。

一、参数控制（辅助方程）

辅助方程的一般形式为

$$\alpha^2(\Delta\boldsymbol{x}^k)^T(\Delta\boldsymbol{x}^k)+\beta^2(\Delta\lambda^k)^2\bar{\boldsymbol{f}}^T\bar{\boldsymbol{f}}=(\Delta l^k)^2 \tag{2.45}$$

上式中左边的两项分别代表了位移和载荷的影响，α 和 β 为尺度因子，不同的取值表征了不同的弧长法。

(1) 球面弧长法(Spherical Arc-Length Method, SALM)

取 $\alpha=1$，$\beta=1$，式(2.45)便是 Crisfield 提出的球面弧长法。此法比较直观，其辅助方程可以看成以 $(\boldsymbol{x}^k,\lambda^k\bar{\boldsymbol{f}})$ 点为球心、以 Δl^k 为半径的球面。由于式中包含了参考荷载向量，尺寸效应明显，一般情况下收敛效果不好。有学者建议采用如下形式的辅助方程，

$$(\Delta\boldsymbol{x}^k)^T(\Delta\boldsymbol{x}^k)+(\Delta\lambda^k)^2=(\Delta l^k)^2 \tag{2.46}$$

显然，上式也没能完全消除尺寸效应的影响。

(2) 柱面弧长法(Cylindrical Arc-Length Method, CALM)

取 $\alpha=1$，$\beta=0$，式(2.45)便是 Crisfield 提出的著名的柱面弧长法。此法忽略了荷载的影响，可以理解为全方位的位移控制方法。由于不存在尺寸效应，许多研究者认为效果较好。

(3) 椭球面弧长法(Ellipsoidal Arc-Length Method, CALM)

K.C. Park 等人提出取 $\alpha=1$，$\beta=S_p$。其中 S_p 为 P.G.Bergan 提出的当前刚度参数，

$$S_p=\frac{(\Delta\boldsymbol{f}^k)^T\Delta\boldsymbol{f}^k}{(\Delta\boldsymbol{f}^k)^T\Delta\boldsymbol{x}^k}$$

这一方法就是椭球面弧长法。椭球面弧长法的优点是不存在尺寸效应。

除上述三种常见方法外，还有 Riks-Wempner-Ramm 二次方法、常数线性化法、沿切平面的 Riks-Wempner 法、沿法平面的 Riks-Wempner 法等，感兴趣的读者可查阅相关文献。

二、弧长增量控制

采用弧长法分析非线性问题时可以采用等弧长增量方法分析，但有时收敛速度很慢，甚至不收敛。因此自动恰当地调整每个增量步中的弧长增量是非常重要的，尤其是在非线性跟踪分析中。而要恰当地做到弧长增量的自动调整，对上一步收敛后的迭代信息进行分析是非常重要的。目前常用的弧长增量控制方式有如下几种：

(1) Crisfield-Bellini 方法

这一方法是 Crisfield 提出的、经 Bellini 修正的方法，其公式为

$$\Delta l^k = \sqrt[4]{N^{k-1}/\bar{N}}\,\Delta l^{k-1} \tag{2.47}$$

式中，$\bar{N}$ 为预先设定的优化迭代次数，一般取 2；N^{k-1} 为上一载荷步的迭代次数，大于 10 时取 10。

(2) Bellini 方法

此方法是由 Bellini 于 1987 年提出的，其公式为

$$\Delta l^k = \sqrt{N^{k-1}/\bar{N}}\,\Delta l^{k-1} \tag{2.48}$$

式中，$\bar{N}$ 一般取 5（当自由度总数小于 25 时）或 6；N^{k-1} 同 Crisfield-Bellini 方法一样处理。

(3) 沈祖炎方法

前述（1）、（2）两种方法仅考虑了上一步收敛所需的迭代次数，沈祖炎等建议采用当前刚度参数增量 ΔS_p^k 及外部控制因子 γ 对 Bellini 的方法进行修正，提出

$$\Delta l^k = \gamma \frac{\Delta \bar{S}_p}{\left|\Delta S_p^k\right|}\sqrt{\frac{N^{k-1}}{\bar{N}}}\,\Delta l^{k-1} \tag{2.49}$$

式中，$\Delta \bar{S}_p$ 为预先设定的当前刚度参数增量，一般取 0.05~0.1；γ 取 0.5~1，非线性程度高时取小值，反之取大值，一般取 0.8 左右。

有学者认为，Bellini 提出的方法具有在载荷平稳变化区间步长变化过于缓慢的缺点，而沈祖炎等提出的方法在接近跳回极值点时步长过小。

三、载荷增量步长符号的确定

由(2.45)式容易求得 $\left|\Delta\lambda^k\right|$，但不能确定 $\Delta\lambda^k$ 的符号，而 $\Delta\lambda^k$ 的符号决定了跟踪分析是向前还是返回，因此非常重要。目前常用的确定规则有四种：①根据当前刚度参数 S_p 的变化确定；②根据增量功的变化确定；③根据当前刚度参数 S_p 的变化和特征向量的分析确定；④根据 $\boldsymbol{K}_T^{k(0)}$ 的行列式值的符号确定。在四种规则中，第一种具有计算简单、几何意义明确的优点，且对大多数问题具有较好的效果。根据这一规则，Riks 提出，

$$\Delta\lambda^k = \begin{cases} \operatorname{sign}(S_p)\left|\Delta\lambda^k\right| & (S_p \neq 0) \\ -\operatorname{sign}(\lambda^{k-1})\left|\Delta\lambda^k\right| & (S_p = 0) \end{cases} \tag{2.50}$$

但这一规则对存在跳回现象的变形过程会发生错误，因此沈祖炎等提出如下形式的规则，

$$\Delta\lambda^k=\begin{cases}(-1)^{nb}\text{sign}(S_p)\left|\Delta\lambda^k\right| & S_p\neq 0\\ -\text{sign}(\Delta\lambda^{k-1})\left|\Delta\lambda^k\right| & S_p=0\end{cases} \tag{2.51}$$

式中，nb为当前步以前出现跳回极值点的个数。

四、载荷增量控制参数求解

根据所采用的辅助方程形式可以给出对应弧长法中荷载增量参数的求解公式。可以看出，球面弧长法、柱面弧长法、椭球面弧长法均需求解关于$\Delta\lambda^k$的一元二次方程，并要求在两个根中作出正确的选择。

$$A\Delta\lambda^{k2}+B\Delta\lambda^k+C=0 \tag{2.52}$$

Crisfield建议按以下标准选择：

① 若$B^2-4AC<0$，减小Δl^0，重新计算。

② 若$B^2-4AC>0$，方程(2.52)式有两根，记为$\Delta\lambda_1^k$和$\Delta\lambda_2^k$，相应的位移增量为$\Delta\boldsymbol{x}_1^k$和$\Delta\boldsymbol{x}_2^k$。分别计算，

$$\beta_1=(\Delta\boldsymbol{x}_1^k+\Delta\lambda_1^k\bar{\boldsymbol{f}})^T(\Delta\boldsymbol{x}^{k-1}+\Delta\lambda^{k-1}\bar{\boldsymbol{f}})$$

$$\beta_2=(\Delta\boldsymbol{x}_2^k+\Delta\lambda_2^k\bar{\boldsymbol{f}})^T(\Delta\boldsymbol{x}^{k-1}+\Delta\lambda^{k-1}\bar{\boldsymbol{f}})$$

③ 取使β为正时所对应的根。如果β_1和β_2同时为正，则取与线性解接近的根。

例 2-1 图 2.6 是一受均匀压力的扁球壳，周边固定。材料为线弹性，$R=100\text{cm}$，$\theta=7.1^\circ$，$E=300\text{GPa}$，$\nu=0.3$。

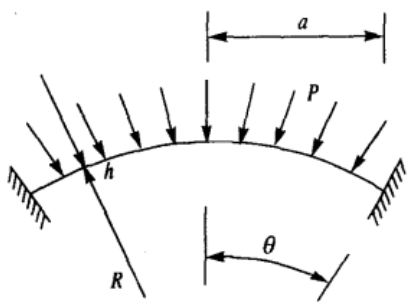


图2.6 均布荷载作用下的扁球壳图

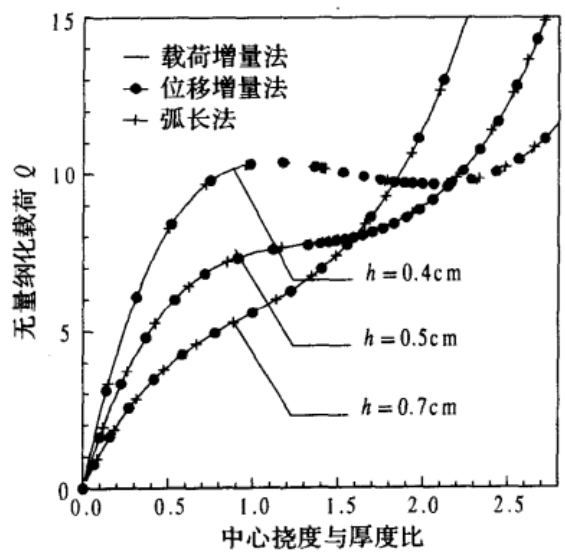


图2.7 扁球壳的载荷与挠度曲线

图2.7给出了不同厚度h时，载荷增量法、位移增量法和弧长法的计算结果。图中，

$$Q=(\frac{a}{h})^4\frac{P}{E}$$

2.3　广义特征值问题的解法

变形体的运动微分方程(2.1)反映了动力问题的两个方面。方程的右端 $\boldsymbol{f}$ 表示外力对变形体的激励，是随时间变化的函数；对于特定的某个变形体的动力问题，它是个已知函数。于是在已知初始条件的情况下可求得该问题的解，即动力响应问题解。该内容将在下一节中讨论。当方程(2.1)的右端为零时， 在非零的初始条件下，该方程也有非零解，此时结构处于自由振动状态。由于没有外载作用，方程的解反映了结构本身的固有特性，这就是结构的固有频率和固有振型，在数学上称为特征值和特征矢量。

求解结构的频率和振型，在工程的实际应用上以及在求解结构动力响应方面，都具有很重要的意义，因而它在结构动力学中占有很重要的地位。

通常，讨论变形体的固有特性时，不计阻尼作用，于是（2.1）式可简化为

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{0} \tag{2.53}$$

假设变形体作简谐振动，即

$$\boldsymbol{x} = \boldsymbol{X}\sin(\omega t + \theta) \tag{2.54}$$

将(2.54)代入（2.53）式，可得如下结构振动的广义特征值问题，

$$\boldsymbol{K}\,\boldsymbol{X} = \omega^2 \boldsymbol{M}\,\boldsymbol{X} \tag{2.55}$$

式中，ω 为圆频率，$\boldsymbol{X}$ 为与时间无关的非零位移矢量。

很显然，由(2.55)求出的圆频率和位移矢量的值， 只取决于变形体的质量矩阵 $\boldsymbol{M}$ 和刚度矩阵 $\boldsymbol{K}$ 。在位移法有限元分析中，$\boldsymbol{M}$ 可以为一致质量阵，也可以为集中质量阵。 对于特征值问题来说，协调单元的集中质量法往往比一致质量法给出较精确的频率。然而，一致质量法可给出结构频率的上限，且能够给出比集中质量法更精确的振型。

对于结构的自由振动问题，由方程（2.55）可得到 N 个特征值（固有频率）及相应的特征矢量（模态、固有振型）。可以证明，特征值和相应的特征矢量具有如下的特性：

(1)如果质量矩阵 $\boldsymbol{M}$ 和刚度矩阵 $\boldsymbol{K}$ 都是对称的，且至少有一个矩阵是正定的，则特征值一定是实数，而特征矢量也可以是实矢量。

(2)如果质量矩阵 $\boldsymbol{M}$ 是正定的，且刚度矩阵 $\boldsymbol{K}$ 为正定（或半正定）的，则所有特征值都是正的（或非负的）的实数。

一般情况下，矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{K}$ 都是正定，或至少是半正定的。因此，可以将所有特征值按大小顺序排列为 $\omega_1 \leqslant \omega_2 \leqslant \cdots \leqslant \omega_N$，对应的特征矢量记为 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_n$。第 i 个特征值和相应特征矢量称为第 i 阶特征对。

（3）显然，特征矢量可以是任意幅度的。为方便起见，当质量矩阵为正定矩阵时，按以下要求规定其幅度，

$$\boldsymbol{\varphi}_i^T \boldsymbol{M} \boldsymbol{\varphi}_i = 1 \qquad (i = 1, 2, \cdots, N) \tag{2.56}$$

上述规定称为特征矢量关于质量矩阵归一化，这样规定的固有振型又称为正则振型。如果没有特别说明，今后所提的振型均为正则振型。

（4）正则振型 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_N$ 关于质量矩阵和刚度矩阵正交，即，

$$\boldsymbol{\varphi}_i^T \boldsymbol{M} \boldsymbol{\varphi}_j = \delta_{ij} \qquad \boldsymbol{\varphi}_i^T \boldsymbol{K} \boldsymbol{\varphi}_j = \delta_{ij}\omega_i^2 \tag{2.57}$$

式中，δ_{ij} 为 Kronecker 符号。因此，正则振型 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_N$ 可以构成 N 维空间的一组基。

由于结构的特征对取决于结构本身，与外部荷载无关，因此特征值和特征矢量具体反映了结构的动力特性。在许多工程中，研究工程结构的动力特性，特别是结构的低阶特征对，就显得十分重要。目前求解特征对的方法很多，常见的有多项式割线刚度法、矢量逆迭代法、广义雅可比法、子空间迭代法和行列式搜索法等。而其中又以子空间迭代法和矢量逆迭代法应用最为广泛。

不失一般性，假定 $\boldsymbol{K}$ 是正定的。由式（2.55）可以导致如下方程

$$\omega^2 = \frac{\boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X}}{\boldsymbol{X}^T \boldsymbol{M} \boldsymbol{X}} \tag{2.58}$$

$$\boldsymbol{X} = \omega^2 \boldsymbol{K}^{-1} \boldsymbol{M} \boldsymbol{X} \tag{2.59}$$

$$\boldsymbol{X} = \frac{1}{\omega^2} \boldsymbol{M}^{-1} \boldsymbol{K} \boldsymbol{X} \tag{2.60}$$

称(2.58)式右端为 Rayleigh 商。由（2.59）式和（2.60）式可以构造两种迭代格式。注意到 ω^2 本身是未知的，为使迭代矢量不至于趋向于零或趋向无穷，迭代过程中必须对迭代矢量进行归一化处理。

$$\boldsymbol{X}^{(n+1)} = \mathrm{Norm}(\boldsymbol{K}^{-1} \boldsymbol{M} \boldsymbol{X}^{(n)}) \tag{2.61}$$

$$\boldsymbol{X}^{(n+1)} = \mathrm{Norm}(\boldsymbol{M}^{-1} \boldsymbol{K} \boldsymbol{X}^{(n)}) \tag{2.62}$$

现在分析(2.61)式和(2.62)式的收敛性。由于初始迭代矢量 $\boldsymbol{X}^{(0)}$ 是任意假定的，它可以分解为

$$\boldsymbol{X}^{(0)} = \sum_{k=1}^{N} \alpha_k \boldsymbol{\varphi}_k \tag{2.63}$$

把（2.63）式代入（2.61）式可得

$$\boldsymbol{X}^{(1)} = \mathrm{Norm}\left(\sum_{k=1}^{N} \alpha_k (\boldsymbol{K}^{-1} \boldsymbol{M} \boldsymbol{\varphi}_k) \right)$$

注意到 $\boldsymbol{M}\boldsymbol{\varphi}_k = \frac{1}{\omega_k^2} \boldsymbol{K}\boldsymbol{\varphi}_k$，上式可化简为

$$\boldsymbol{X}^{(1)} = \mathrm{Norm}\left(\sum_{k=1}^{N} \frac{\alpha_k}{\omega_k^2} \boldsymbol{\varphi}_k \right) \tag{2.64}$$

上式说明，相对 $\boldsymbol{X}^{(0)}$，$\boldsymbol{X}^{(1)}$ 中频率高的模态分量大大减少了。因此，迭代格式（2.61）将收敛到频率最小的模态。当频率比较密集时，这种迭代格式收敛很慢。同样，可以证明迭代格式（2.62）收敛于高阶模态。通常我们更关心结构的低阶固有模态和固有频率，利用迭代格式（2.61）可以得到两种常用的广义特征值迭代方法：矢量逆迭代法和子空间迭代法。

2.3.1 矢量逆迭代法

用矢量逆迭代法计算低阶特征对是很有效的。假定初始迭代矢量为 $\boldsymbol{X}^{(0)}$（其 N 个分量一般采用随机数产生），作

$$\boldsymbol{Y}^{(0)} = \boldsymbol{M}\ \boldsymbol{X}^{(0)} \tag{2.65}$$

对 $k = 0, 1, 2, \cdots$，计算，

$$\boldsymbol{X}^{(k+1)} = \boldsymbol{K}^{-1}\boldsymbol{Y}^{(k)} \tag{2.66}$$

$$\boldsymbol{Y}^{(k+1)} = \boldsymbol{M}\boldsymbol{X}^{(k+1)} \tag{2.67}$$

$$\rho^{(k+1)} = \frac{(\boldsymbol{X}^{(k+1)})^T\boldsymbol{Y}^{(k)}}{(\boldsymbol{X}^{(k+1)})^T\boldsymbol{Y}^{(k+1)}} = \frac{(\boldsymbol{X}^{(k+1)})^T\boldsymbol{K}\boldsymbol{X}^{(k+1)}}{(\boldsymbol{X}^{(k+1)})^T\boldsymbol{M}\boldsymbol{X}^{(k+1)}} \tag{2.68}$$

$$\boldsymbol{Y}^{(k+1)} = \frac{\boldsymbol{Y}^{(k+1)}}{\left(\boldsymbol{X}^{(k+1)T}\boldsymbol{Y}^{(k+1)}\right)^{1/2}} = \boldsymbol{M}\frac{\boldsymbol{X}^{(k+1)}}{\boldsymbol{X}^{(k+1)}\boldsymbol{M}\boldsymbol{X}^{(k+1)}} \tag{2.69}$$

可以证明，只要 $\boldsymbol{\varphi}_1^T\boldsymbol{M}\boldsymbol{X}^{(0)} \neq 0$，则有

$$\lim_{k\to\infty}\boldsymbol{Y}^{(k+1)} = \boldsymbol{M}\boldsymbol{\varphi}_1 \tag{2.70}$$

$$\lim_{k\to\infty}\rho^{(k+1)} = \omega_1^2 \tag{2.71}$$

迭代过程可以采用下式进行收敛性判定，

$$\left|\frac{\rho^{(k+1)} - \rho^{(k)}}{\rho^{(k+1)}}\right| \leq \beta \tag{2.72}$$

式中，β 为事先设定的容许误差。采用上述方法可以求得基频及相应模态。

现考虑非基频及相应模态的分析，假设 m 个特征矢量 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_m$ 已经求得，要求第 m+1 个特征值和特征矢量。设 $\boldsymbol{X}^{(k)}$ 为第 k 步迭代矢量，一般来说，$\boldsymbol{X}^{(k)}$ 包含 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_m$ 的成分。为使迭代收敛于第 m+1 阶特征矢量，迭代过程中必须不断对 $\boldsymbol{X}^{(k)}$ 进行 Gram-Schmidt 正交化处理，即

$$\overline{\boldsymbol{X}}^{(k)} = \boldsymbol{X}^{(k)} - \sum_{i=1}^{m}\alpha_i\boldsymbol{\varphi}_i \tag{2.73}$$

$$\alpha_i = \boldsymbol{\varphi}_i^T\boldsymbol{M}\boldsymbol{X}^{(k)} \tag{2.74}$$

以使迭代收敛于第 m+1 阶特征值和特征矢量。

2.3.2　子空间迭代法

子空间迭代法是一种求解大型特征值问题低阶特征对很有效的方法。它的特点是利用瑞利-里兹变换，将高阶方程投影到一个低维空间（即子空间）中，然后在子空间内求解一个低阶的广义特征方程，并以求出的低阶特征矢量返回到原方程的正交里兹(Ritz)基矢量，然后以逆迭代的形式修正里兹基，使其构成的子空间接近于原方程中所要的低阶特征矢量构成的子空间，原方程在这个近似的低维空间中就能求出近似的低阶特征对。整个过程就是在矢量的迭代和子空间内求低阶广义特征方程这两个方面交替进行，反复迭代而不断逼近真实解。

子空间迭代法的基本步骤如下：

设要求 N 阶广义特征值问题的前 p 阶（p<N）特征对，先假定 q(q>p)个线性无关的初始迭代矢量(即里兹基)，并将这 q 列矢量依次排列构成一个 n×q 阶的矩阵 $\boldsymbol{A}^{(0)}$，开始对 $k = 0, 1, 2, \cdots$，按如下格式循环迭代。

（1）矢量逆迭代求出 $\overline{\boldsymbol{A}}^{(k+1)}$

$$\overline{\boldsymbol{A}}^{(k+1)} = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{A}^{(k)} \tag{2.75}$$

（2）将原方程投影到子空间中

$$\boldsymbol{K}^{(k+1)} = (\bar{\boldsymbol{A}}^{(k+1)})^T \boldsymbol{K}\bar{\boldsymbol{A}}^{(k+1)} \tag{2.76}$$

$$\boldsymbol{M}^{(k+1)} = (\bar{\boldsymbol{A}}^{(k+1)})^T \boldsymbol{M}\bar{\boldsymbol{A}}^{(k+1)} \tag{2.77}$$

（3）求解子空间内的特征方程

$$\boldsymbol{K}^{(k+1)}\boldsymbol{B}^{(k+1)} = \boldsymbol{M}^{(k+1)}\boldsymbol{\Lambda}^{(k+1)}\boldsymbol{B}^{(k+1)} \tag{2.78}$$

式中，$\boldsymbol{B}^{(k+1)}$ 和 $\boldsymbol{\Lambda}^{(k+1)}$ 分别为子空间中特征矢量和特征值构成的矩阵，$\boldsymbol{\Lambda}^{(k+1)}$ 为对角矩阵。

（4）计算改进后的特征矢量

$$\boldsymbol{A}^{(k+1)} = \bar{\boldsymbol{A}}^{(k+1)}\ \boldsymbol{B}^{(k+1)} \tag{2.79}$$

可以证明，只要 $\boldsymbol{A}^{(0)}$ 的各列矢量不与方程的最低的 q 阶特征矢量正交，一定有

$$\lim_{k\to\infty} \boldsymbol{\Lambda}^{(k+1)} = \boldsymbol{\Lambda} \tag{2.80}$$

$$\lim_{k\to\infty} \boldsymbol{A}^{(k+1)} = \boldsymbol{A} \tag{2.81}$$

式中，$\boldsymbol{A}$ 和 $\boldsymbol{\Lambda}$ 分别为广义特征值问题（2.55）的前 q 阶模态和特征值构成的矩阵。迭代过程可以采用类似矢量逆迭代法的判断标准判断误差。q 一般按下式选取，

$$q = \min(p+8, N) \tag{2.82}$$

2.3.3 移频技术

如果刚度矩阵 $\boldsymbol{K}$ 是一个奇异矩阵，其逆矩阵不存在，则矢量逆迭代法和子空间迭代法失效。移频技术是解决这一问题最简单而且有效的方法。将(2.55)式两端同时加上 $\lambda\boldsymbol{MX}$ 得，

$$\hat{\boldsymbol{K}}\boldsymbol{X} = \hat{\omega}^2\ \boldsymbol{M}\ \boldsymbol{X} \tag{2.83}$$

式中，$\hat{\boldsymbol{K}} = \boldsymbol{K} + \lambda\boldsymbol{M}$，$\hat{\omega}^2 = \lambda + \omega^2$。当 $\lambda \neq 0$ 时，一般可以保证 $\hat{\boldsymbol{K}}$ 是非奇异的。(2.83)可以看成一新结构的自由振动问题，该结构与原结构具有相同的固有模态，其固有频率 $\hat{\omega}_k$ ($k = 1, 2, \ldots, N$) 和原结构的固有频率间满足如下关系，

$$\hat{\omega}_k^2 = \omega_k^2 + \lambda \tag{2.84}$$

当 $\lambda \approx -\omega_0^2$ 时，矢量逆迭代法首先得到的是频率最接近 ω_0 的那个特征对，这一点在实际工程中也是非常有用的。譬如，斜拉桥自由振动分析时，斜拉索局部自由振动的固有频率往往比整桥自由振动的固有频率低，直接采用矢量逆迭代法很难求得整桥的自由振动特性，恰当地选择 ω_0 则可以方便地得到我们关心的固有频率和模态。

例 2-2 考虑一个两自由度系统，其自由振动的运动方程为

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} 6 & -2 \\ -2 & 4 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

采用矢量逆迭代法求该系统的固有频率和固有振型。随机初值条件下，一阶和二阶特征对迭代结果如表 2.1 所示。根据一般的线性代数方法，其解答为

$$\omega_1^2 = 2 \qquad \boldsymbol{\varphi}_1 = \left\{\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right\}^T$$

$$\omega_2^2 = 5 \qquad \varphi_2 = \left\{\frac{1}{2}\sqrt{\frac{2}{3}},\ -\sqrt{\frac{2}{3}}\right\}^T$$

比较可见，两者吻合得很好。

表 2.1　矢量逆迭代法计算固有频率和固有振型

迭代次数	1	2	3	4	5	6	7	8
φ_1	0.38666	0.50644	0.55040	0.56683	0.57319	0.57569	0.57669	0.57709
	0.83725	0.69788	0.62778	0.59783	0.58559	0.58065	0.57867	0.57788
ω_1^2	2.40606	2.07330	2.01197	2.00192	2.00031	2.00005	2.00001	2.00000
φ_2	-0.40825	-0.40825	-0.40825					
	0.81650	0.81650	0.81650					
ω_2^2	5.00002	5.00000	5.00000					

例 2-3　图 2.8 为一三边自由、一边固定的悬臂方板，边长 $a=2$m，板厚 $h=0.1$m。采用第 7 章介绍的实体退化单元离散，如图示划分为 100 个单元，共 682 个节点。材料弹性模量为 $E=2.06\times10^5$MPa，泊松比 $\nu=0.3$，质量密度为 $\rho=7800\text{kg}/\text{m}^3$。采用矢量逆迭代法求得结构前 9 阶特征对，式中第 6 阶特征对反映悬臂方板的面内振动。定义

$$\varsigma=\omega a^2\sqrt{\rho h/D}$$

式中，D 为板的弯曲刚度。则悬臂方板前 8 阶横向振动的固有频率对应的 ς 值依次为：3.46202，8.36127，21.0415，26.7169，30.2236，52.1977，59.8318，62.2609，固有振型挠度等高线如图 2.9 示。

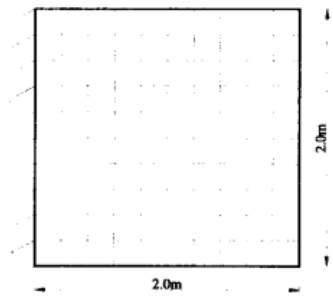


图 2.8　悬臂方板的有限元网格

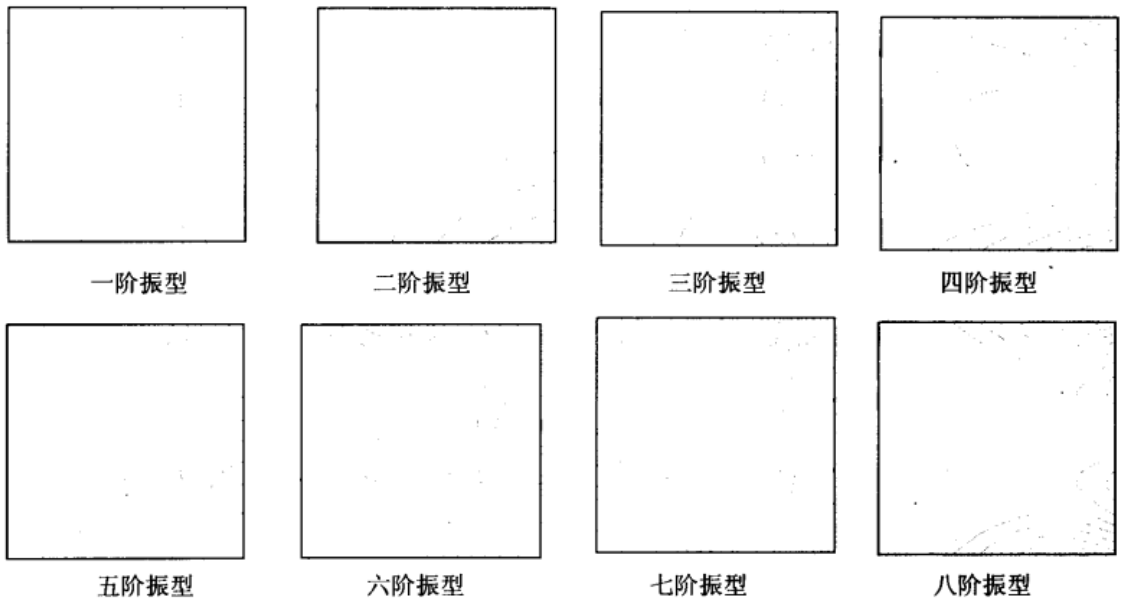


图 2.9　前八阶横向振动振型示意图

2.4 动力方程的解法

方程（2.1）是一组关于时间 t 的二阶微分方程，这一方程的求解方法可以分为两大类，即振型叠加法和直接积分法。振型迭加法是用结构的振型（特征矢量）来形成里兹基，通过对方程（2.1）作变换，用广义坐标代替原来的有限元节点位移坐标，以降低方程的阶数。由于固有振型的正交性，振型迭加法不仅可以降低阶数，而且可以解除原方程中各自由度间的耦合，简化了方程的运算。一般地说，在外力作用下结构通常只激起较低的一部分振型，而大多数高阶振型分量很小，可以忽略不计。对 N 阶动力方程，起主要作用的是其前 q 阶振型，且 $q << N$。因此模态迭加法的关键是确定 q 和模态分析，而合理准确地确定 q 值是比较困难的。

当外载荷为冲击荷载，激起振型很多时，采用直接积分法就比振型迭加法更加合适。对非线性动力响应问题，则只能采用直接积分法。直接积分法是将动力方程在时间域上离散，并作近似的插值，化为差分格式，然后根据初始条件，利用离散后导出的线性代数方程组逐步求解在各离散时刻上的结构响应。在时间域上作不同的插值处理，就可得到各种不同的直接积分公式。目前，在工程中和计算机程序中最常用的有如下几种方法：中央差分法、Houbolt 法、Wilson-θ 法和 Newmark 法等，而其中又以 Wilson-θ 法和 Newmark 法应用最广。精细时程积分法是钟万勰院士提出的一种高精度和高效率求解方法。

2.4.1 振型叠加法

振型叠加法是在是在积分运动方程前，利用系统自由振动的固有振型将方程组转化为 q（$q \le N$）个互不耦合的方程，然后对每一个方程解析或进行数值积分。当采用数值积分时，每个方程可根据其固有频率选择不同的时间步长。振型叠加法可分成以下三大步骤。

一、求解系统的固有频率和固有模态

利用上一节介绍的方法，可以求出系统的前 q 阶固有频率 $\omega_1,\omega_2,...,\omega_q$ 和正则模态 $\varphi_1,\varphi_2,...,\varphi_q$。

二、求在子空间中响应

将正则模态 $\varphi_1,\varphi_2,...,\varphi_q$ 写成矩阵形式 $\boldsymbol{A}=\left[\varphi_1,\varphi_2,...,\varphi_q\right]$。假设(2.1)式中，荷载激起响应主要以前 q 阶低频为主，高频分量很少、可以忽略不计，则

$$\boldsymbol{x}=\boldsymbol{A}\boldsymbol{\zeta} \tag{2.85}$$

将上式代入(2.1)式，方程两边同时乘以 $\boldsymbol{A}^T$，利用正则振型的特性有

$$\ddot{\boldsymbol{\zeta}}+\hat{\boldsymbol{C}}\dot{\boldsymbol{\zeta}}+\hat{\boldsymbol{K}}\boldsymbol{\zeta}=\hat{\boldsymbol{f}} \tag{2.86}$$

式中，$\hat{\boldsymbol{C}}=\boldsymbol{A}^T\boldsymbol{C}\,\boldsymbol{A}$，$\hat{\boldsymbol{K}}=\mathrm{diag}\{\omega_1^2,\omega_2^2,...,\omega_q^2\}$，$\hat{\boldsymbol{f}}=\boldsymbol{A}^T\boldsymbol{f}$。若阻尼矩阵是瑞利（Rayleigh）阻尼，即 $\boldsymbol{C}$ 满足，

$$\boldsymbol{C}=\alpha\boldsymbol{M}+\beta\boldsymbol{K} \tag{2.87}$$

则 $\hat{\boldsymbol{C}}$ 与 $\hat{\boldsymbol{K}}$ 同为对角矩阵，式(2.86)完全解耦，原问题简化为 q 个独立的单自由度问题，可以采用解析或数值的方法求解。

三、振型叠加得到系统的响应

将子空间中求得的前 q 阶振型响应，代回到(2.85)式将它们叠加起来，就得到系统的近似

响应，也就是每个节点的位移值。

由以上计算过程可以看出，振型叠加法具有一定的局限性。首先，振型叠加法只适用于线性系统的分析，对于非线性系统，叠加原理不再适用。其次，该方法也不适合分析局部受高频冲击荷载作用的系统响应，因为此时系统响应的高频分量不可以忽略，为保证有足够高的计算精度，q 必须取得很大，计算系统的固有振型将花费大量的计算机时间。第三，阻尼矩阵最好具有如下性质：振型关于阻尼矩阵正交。

2.4.2　Wilson-θ 法

假定 $t=0$ 时，结构的位移 $\boldsymbol{x}^0$ 和速度 $\dot{\boldsymbol{x}}^0$，由(2.1)式可以求出初始时刻的加速度 $\ddot{\boldsymbol{x}}^0$。将求解的时间区域划分成 n 份，记时间分点序列为 $0=t_0<t_1<t_2<\ldots<t_n=T$，记 $\Delta t_k=t_{k+1}-t_k$，$k=0,1,\cdots,n$。

Wilson-θ法是线性加速度法的推广。线性加速度法是假定在 $[t_k,t_{k+1}]$ 时间内，即在步长 Δt_k 时间内，加速度 $\ddot{\boldsymbol{x}}$ 是线性变化，其表达式为

$$\ddot{\boldsymbol{x}}=\ddot{\boldsymbol{x}}^k+\frac{\tau}{\Delta t_k}(\ddot{\boldsymbol{x}}^{k+1}-\ddot{\boldsymbol{x}}^k) \tag{2.88}$$

式中，$\tau\in[0,\Delta t_k]$。从此假设出发就可以导出线性加速度的递推公式。但是由算法稳定性分析得知线性加速度法是条件稳定的。Wilson-θ法推广了线性加速度法，认为加速度在 $[t_k,t_k+\theta\Delta t_k]$ 时间内是线性变化的，即

$$\ddot{\boldsymbol{x}}=\ddot{\boldsymbol{x}}^k+\frac{\tau}{\theta\Delta t_k}\left(\ddot{\boldsymbol{x}}^{k+\theta}-\ddot{\boldsymbol{x}}^k\right) \tag{2.89}$$

这里 $\theta\geq 1$，$\ddot{\boldsymbol{x}}^{k+\theta}$ 为 $t_k+\theta\Delta t_k$ 时刻的加速度。可以证明当 $\theta\geq 1.37$ 时，Wilson-θ 法是无条件稳定的，通常取 $\theta=1.40$。积分上式可得

$$\dot{\boldsymbol{x}}=\dot{\boldsymbol{x}}^k+\ddot{\boldsymbol{x}}^k\cdot\tau+\frac{\tau^2}{2\theta\Delta t_k}(\ddot{\boldsymbol{x}}^{k+\theta}-\ddot{\boldsymbol{x}}^k) \tag{2.90}$$

$$\boldsymbol{x}=\boldsymbol{x}^k+\dot{\boldsymbol{x}}^k\cdot\tau+\frac{1}{2}\ddot{\boldsymbol{x}}^k\tau^2+\frac{1}{6\theta\Delta t_k}\tau^3\left(\ddot{\boldsymbol{x}}^{k+\theta}-\ddot{\boldsymbol{x}}^k\right) \tag{2.91}$$

令上述两式中 $\tau=\theta\Delta t_k$，可以解出 $t_k+\theta\Delta t_k$ 时刻的速度和加速度。即，

$$\ddot{\boldsymbol{x}}^{k+\theta}=\frac{6}{\theta^2{\Delta t_k}^2}\left(\boldsymbol{x}^{k+\theta}-\boldsymbol{x}^k\right)-\frac{6}{\theta\Delta t_k}\dot{\boldsymbol{x}}^k-2\ddot{\boldsymbol{x}}^k \tag{2.92}$$

$$\dot{\boldsymbol{x}}^{k+\theta}=\frac{3}{\theta\Delta t_k}\left(\boldsymbol{x}^{k+\theta}-\boldsymbol{x}^k\right)-2\dot{\boldsymbol{x}}^k-\frac{\theta\Delta t_k}{2}\ddot{\boldsymbol{x}}^k \tag{2.93}$$

代入方程（2.1）可以得到 $t_k+\theta\Delta t_k$ 时刻的运动方程，

$$\bar{\boldsymbol{K}}\,\boldsymbol{x}^{k+\theta}=\bar{\boldsymbol{f}} \tag{2.94}$$

式中，

$$\bar{\boldsymbol{K}}=\boldsymbol{K}+\frac{6}{\theta^2\Delta t_k^2}\boldsymbol{M}+\frac{3}{\theta\Delta t_k}\boldsymbol{C} \tag{2.95}$$

$$\begin{aligned}\bar{\boldsymbol{f}} &= \boldsymbol{f}^k + \theta(\boldsymbol{f}^{k+1} - \boldsymbol{f}^k) + \boldsymbol{M}\left(\frac{6}{\theta^2\Delta t_k^2}\boldsymbol{x}^k + \frac{6}{\theta\Delta t_k}\dot{\boldsymbol{x}}^k + 2\ddot{\boldsymbol{x}}^k\right) \\ &+ \boldsymbol{C}\left(\frac{3}{\theta\Delta t_k}\boldsymbol{x}^k + 2\dot{\boldsymbol{x}}^k + \frac{\theta\Delta t_k}{2}\ddot{\boldsymbol{x}}^k\right)\end{aligned} \tag{2.96}$$

注意上式中，$t_k + \theta\Delta t_k$ 时刻的 $\boldsymbol{f}$ 是由 $\boldsymbol{f}^k$ 和 $\boldsymbol{f}^{k+1}$ 线性插值得到的。由(2.94)式求出 $\boldsymbol{x}^{k+\theta}$ 后代入（2.89）、（2.90）和(2.91)式，并令 $\tau = \Delta t_k$，可以求得

$$\ddot{\boldsymbol{x}}^{k+1} = \frac{6}{\theta^2\Delta t_k^2}(\boldsymbol{x}^{k+\theta} - \boldsymbol{x}^k) - \frac{6}{\theta\Delta t_k}\dot{\boldsymbol{x}}^k + (1 - \frac{3}{\theta})\ddot{\boldsymbol{x}}^k \tag{2.97}$$

$$\dot{\boldsymbol{x}}^{k+1} = \dot{\boldsymbol{x}}^k + \frac{\Delta t_k}{2}(\ddot{\boldsymbol{x}}^{k+1} + \ddot{\boldsymbol{x}}^k) \tag{2.98}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta t_k\dot{\boldsymbol{x}}^k + \frac{\Delta t_k^2}{6}(\ddot{\boldsymbol{x}}^{k+1} + 2\ddot{\boldsymbol{x}}^k) \tag{2.99}$$

2.4.3 Newmark 法

Newmark 法是由 Newmark 于 1959 年提出的，其基本假定是：

$$\dot{\boldsymbol{x}}^{k+1} = \dot{\boldsymbol{x}}^k + [(1-\delta)\ddot{\boldsymbol{x}}^k + \delta\,\ddot{\boldsymbol{x}}^{k+1}]\Delta t_k \tag{2.100}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \dot{\boldsymbol{x}}^k\Delta t_k + \left[\left(\frac{1}{2} - \alpha\right)\ddot{\boldsymbol{x}}^k + \alpha\ddot{\boldsymbol{x}}^{k+1}\right]\Delta t_k^2 \tag{2.101}$$

式中，α 和 δ 是按积分的精度和稳定性要求可调整的参数。

当 $\delta = 1/2$，α=1/6 时，Newmark 法退化为线性加速度法。当 $\delta = 1/2$，α=1/4 时，Newmark 法退化为平均加速度法，是目前用得很广泛的逐步积分法。

由（2.100）式和(2.101)式不难求出，

$$\ddot{\boldsymbol{x}}^{k+1} = \frac{1}{\alpha\Delta t_k^2}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) - \frac{1}{\alpha\Delta t_k}\dot{\boldsymbol{x}}^k - \left(\frac{1}{2\alpha} - 1\right)\ddot{\boldsymbol{x}}^k \tag{2.102}$$

$$\dot{\boldsymbol{x}}^{k+1} = \frac{\delta}{\alpha\Delta t_k}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) + \left(1 - \frac{\delta}{\alpha}\right)\dot{\boldsymbol{x}}^k + \left(1 + \frac{\delta}{2\alpha}\right)\Delta t_k\ddot{\boldsymbol{x}}^k \tag{2.103}$$

代入（2.1）式可得

$$\bar{\boldsymbol{K}}\,\boldsymbol{x}^{k+1} = \bar{\boldsymbol{f}} \tag{2.104}$$

式中，

$$\bar{\boldsymbol{K}} = \boldsymbol{K} + \frac{1}{\alpha\Delta t_k^2}\boldsymbol{M} + \frac{\delta}{\alpha\Delta t_k}\boldsymbol{C} \tag{2.105}$$

$$\begin{aligned}\bar{\boldsymbol{f}} &= \boldsymbol{f}^{k+1} + \boldsymbol{M}\left[\frac{1}{\alpha\Delta t_k}\dot{\boldsymbol{x}}^k + \left(\frac{1}{2\alpha} - 1\right)\ddot{\boldsymbol{x}}^k\right] \\ &+ \boldsymbol{C}\left[\frac{\delta}{\alpha\Delta t_k}\boldsymbol{x}^k + \left(\frac{\delta}{\alpha} - 1\right)\dot{\boldsymbol{x}}^k + \left(\frac{\delta}{2\alpha} - 1\right)\Delta t_k\ddot{\boldsymbol{x}}^k\right]\end{aligned} \tag{2.106}$$

求解方程即可得到 t_{k+1} 时刻的位移，代入（2.102）式和（2.103）式即可求得 t_{k+1} 时刻的速度和加速度。

数值稳定性分析表明，当$\delta \geqslant 0.5$、$\alpha \geqslant 0.25(0.5+\delta)^2$时，Newmark 法是无条件稳定的。在实际应用中应当注意，当δ不取 1/2 时，Newmark 法将引入虚假阻尼，且与$(\delta-\frac{1}{2})$成比例。当$\delta<\frac{1}{2}$时，导致负阻尼，于是数值方法本身会产生虚假的自激振动；$\delta>\frac{1}{2}$时，导致正阻尼，使得响应幅值衰减。即使取$\delta=\frac{1}{2}$、$\alpha=\frac{1}{4}$，在频响分析中也可能会导致较大的截断误差，从而使多自由度系统的响应峰值不准确。

对于非线性动力学问题，求解时往往在一个时间步内采用 Newton-Raphson 法或修正的 Newton-Raphson 法进行迭代分析。当一个时间步内刚度矩阵变化不大时，每一时间步一般只需修正一次刚度阵。

2.4.4 精细时程积分法

若初值问题(2.1)中$\boldsymbol{M}$、$\boldsymbol{K}$、$\boldsymbol{C}$均为常数矩阵，$\boldsymbol{f}$只是时间的函数，则(2.1)即为线性系统的初值问题。经过适当变化，(2.1)式可表示为如下一阶初值问题：

$$\dot{\boldsymbol{V}}=\boldsymbol{H}\boldsymbol{V}+\boldsymbol{r} \tag{2.107}$$

式中，

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{D}\\ \boldsymbol{B} & \boldsymbol{G}\end{bmatrix} \qquad \boldsymbol{V}=\begin{Bmatrix}\boldsymbol{q}\\ \boldsymbol{p}\end{Bmatrix} \qquad \boldsymbol{r}=\begin{Bmatrix}\boldsymbol{0}\\ \boldsymbol{f}\end{Bmatrix}$$

其中，$\boldsymbol{q}=\boldsymbol{x}$，$\boldsymbol{p}=\boldsymbol{M}\dot{\boldsymbol{x}}+\boldsymbol{C}\boldsymbol{x}/2$，$\boldsymbol{A}=-\boldsymbol{M}^{-1}\boldsymbol{C}/2$，$\boldsymbol{B}=\boldsymbol{C}\boldsymbol{M}^{-1}\boldsymbol{C}/4-\boldsymbol{K}$，$\boldsymbol{G}=-\boldsymbol{C}\boldsymbol{M}^{-1}/2$，$\boldsymbol{D}=\boldsymbol{M}^{-1}$。设第$k$时间步的$\boldsymbol{V}$值已知，记为$\boldsymbol{V}^k$，则$\boldsymbol{V}^{k+1}$可表示为：

$$\boldsymbol{V}^{k+1}=\boldsymbol{T}\left[\boldsymbol{V}^k+\boldsymbol{H}^{-1}\left(\boldsymbol{r}_0+\boldsymbol{H}^{-1}\boldsymbol{r}_1\right)\right]-\boldsymbol{H}^{-1}\left[\boldsymbol{r}_0+\boldsymbol{H}^{-1}\boldsymbol{r}_1+\boldsymbol{r}_1\Delta t_k\right] \tag{2.108}$$

式中，$\boldsymbol{r}_0=\boldsymbol{r}^k$，$\boldsymbol{r}_1=\dfrac{\boldsymbol{r}^{k+1}-\boldsymbol{r}^k}{\Delta t_k}$，$\boldsymbol{T}=\mathrm{e}^{\boldsymbol{H}\Delta t_k}$。精细矩阵$\boldsymbol{T}$计算如下：

$$\boldsymbol{T}=\mathrm{e}^{\boldsymbol{H}\Delta t_k}=(\mathrm{e}^{\boldsymbol{H}\Delta t_k/m})^m \tag{2.109}$$

取$m=2^M$，$\Delta\tau_k=\Delta t_k/m$。若$\Delta\tau_k$足够小，则有

$$\mathrm{e}^{(\boldsymbol{H}\Delta\tau_k)}=\boldsymbol{I}+\boldsymbol{H}\Delta\tau_k+(\boldsymbol{H}\Delta\tau_k)^2/2=\boldsymbol{I}+\boldsymbol{T}_\alpha \tag{2.110}$$

其中，$\boldsymbol{T}_\alpha=(\boldsymbol{H}\Delta\tau_k)\left[\boldsymbol{I}+(\boldsymbol{H}\Delta\tau_k)/2\right]$。将(2.110)代入(2.109)式得：

$$\boldsymbol{T}=\left[\mathrm{e}^{(\boldsymbol{H}\Delta\tau_k)}\right]^{2^M}=\left[\boldsymbol{I}+\boldsymbol{T}_\alpha\right]^{2^M}=\left[\boldsymbol{I}+\boldsymbol{T}_\alpha\right]^{2^{M-1}}\left[\boldsymbol{I}+\boldsymbol{T}_\alpha\right]^{2^{M-1}} \tag{2.111}$$

而且，$(\boldsymbol{I}+\boldsymbol{T}_\alpha)\times(\boldsymbol{I}+\boldsymbol{T}_\alpha)=\boldsymbol{I}+2\boldsymbol{T}_\alpha+\boldsymbol{T}_\alpha\times\boldsymbol{T}_\alpha$。计算$\boldsymbol{T}$的过程相当于在程序中执行

$\boldsymbol{T}_\alpha=(\boldsymbol{H}\Delta\tau_k)\left[\boldsymbol{I}+(\boldsymbol{H}\Delta\tau_k)/2\right]$

DO　$i=0, M-1$

　　$\boldsymbol{T}_\alpha=2\boldsymbol{T}_\alpha+\boldsymbol{T}_\alpha\times\boldsymbol{T}_\alpha$

END　DO

$\boldsymbol{T}=\boldsymbol{T}_\alpha+\boldsymbol{I}$

一旦求出矩阵 $\boldsymbol{T}$，由初始条件按(2.108)式可以计算任意时刻的响应。

例 2-4 考虑例 2-2 系统，其运动方程为

$$\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}\begin{Bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{Bmatrix}+\begin{bmatrix}6 & -2\\ -2 & 4\end{bmatrix}\begin{Bmatrix}x_1\\ x_2\end{Bmatrix}=\begin{Bmatrix}0\\ 10\end{Bmatrix}$$

初始条件为：$t=0$，$\boldsymbol{x}_0=\boldsymbol{0}$，$\dot{\boldsymbol{x}}_0=\boldsymbol{0}$。利用例 2-2 求得的正则振型，将原问题转换为以 $\boldsymbol{\varphi}_1$，$\boldsymbol{\varphi}_2$ 为基向量的两个互不耦合的运动方程，

$$\ddot{\zeta}_1(t)+2\zeta(t)=\frac{10}{\sqrt{3}}\qquad \ddot{\zeta}_1(t)+2\zeta(t)=\frac{10}{\sqrt{3}}$$

原问题的初始条件也可转换为 $\zeta_1=\zeta_2=0$，$\dot{\zeta}_1=\dot{\zeta}_2=0$，。利用无阻尼条件下的杜哈美积分可得到上式的精确解，

$$\zeta_1(t)=\frac{5}{\sqrt{3}}(1-\cos\sqrt{2}t)\qquad \zeta_2(t)=2\sqrt{\frac{2}{3}}(-1+\cos\sqrt{5}t)$$

最后利用振型叠加得到原系统的解析解，

$$\begin{Bmatrix}x_1(t)\\ x_2(t)\end{Bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{3}} & \frac{1}{2}\sqrt{\frac{2}{3}}\\ \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}}\end{bmatrix}\begin{Bmatrix}\frac{5}{\sqrt{3}}(1-\cos\sqrt{2}t)\\ 2\sqrt{\frac{2}{3}}(-1+\cos\sqrt{5}t)\end{Bmatrix}$$

采用 Wilson-θ 法、Newmark 法和精细时程积分法也可以求得该问题的近似解，表 2.2 给出了时间步长 $\Delta t=0.28$ 时四种方法的解。其中模态叠加法为精确解，精细时程积分法中取 $M=1$。由表可以看出，精细时程积分法也给出了精确的数值解。

表 2.2 不同方法的计算结果比较

时 间		Δt	$2\Delta t$	$3\Delta t$	$4\Delta t$	$5\Delta t$	$6\Delta t$	$7\Delta t$	$8\Delta t$	$9\Delta t$
模 态 叠加法	x_1	0.003	0.038	0.176	0.486	0.996	1.657	2.338	2.861	3.052
	x_2	0.382	1.412	2.781	4.094	4.996	5.291	4.986	4.277	3.457
Wilson-θ 法	x_1	0.006	0.053	0.196	0.490	0.952	1.542	2.162	2.670	2.923
	x_2	0.366	1.339	2.639	3.924	4.879	5.309	5.178	4.606	3.818
Newmark 法	x_1	0.007	0.050	0.189	0.485	0.961	1.581	2.233	2.761	3.004
	x_2	0.364	1.351	2.683	3.995	4.950	5.337	5.130	4.478	3.642
精细时程积分法	x_1	0.003	0.038	0.176	0.486	0.996	1.657	2.338	2.861	3.052
	x_2	0.382	1.412	2.781	4.094	4.996	5.291	4.986	4.277	3.458

2.5 数值积分

在有限元分析中，经常需要精确并快速地计算如下形式的积分，

$$\int_{-1}^{1}f(\xi)\mathrm{d}\xi\quad \int_{-1}^{1}\int_{-1}^{1}f(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta\quad \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}f(\xi,\eta,\zeta)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{2.112}$$

$$\int_{0}^{1}\int_{0}^{1-L_1}f(L_1,L_2)\mathrm{d}L_2\mathrm{d}L_1\quad \int_{0}^{1}\int_{0}^{1-L_1}\int_{0}^{1-L_1-L_2}f(L_1,L_2,L_3)\mathrm{d}L_3\mathrm{d}L_2\mathrm{d}L_1 \tag{2.113}$$

一般来说，式中的被积函数很复杂，很难给出它的解析表达式，因此有限元法一般都采用数

值积分。

数值积分是一种利用被积函数在有限个点上的函数值来推算积分近似值的有效方法，以一维积分为例，数值积分公式的基本形式是：

$$\int_{-1}^{1} f(\xi)\mathrm{d}\xi \approx \sum_{k=0}^{n} \omega_k f(\xi_k) \tag{2.114}$$

式中，ξ_k 称为求积节点，ω_k 为求积系数，n 为求积节点数。

2.5.1　Gauss 积分

首先讨论一维问题的数值积分 $\int_{-1}^{1} f(\xi)\mathrm{d}\xi$，对于 n 个积分点，按照积分点位置选取不同，通常可采用两种不同的数值积分方法，即 Newton-Cotes 积分和 Gauss 积分。Newton-Cotes 积分方法采用包括积分端点在内的等间距分布求积节点，而 Gauss 积分方法则按最高代数精度阶的要求来确定积分点位置。因此，在求积节点数 n 相同的情况下，Gauss 积分法比 Newton-Cotes 积分法具有更高的代数精度阶，有限元分析一般均采用 Gauss 积分。

当 $n \le 6$ 时，Gauss 积分的求积节点和求积系数可以从表 2.3 中查到。求积节点数目的选取与被积函数 $f(\xi)$ 有关。当 $f(\xi)$ 是 m 次多项式时，则取 $n \ge (m+1)/2$。当 $f(\xi)$ 不是多项式时，则需要通过试算来判断选取恰当的 n 值。须指出的是，n 不能取得过大，因为计算工作量会随求积节点数增加而急剧增加。

逐次利用（2.114）式，则有以下二维和三维 Gauss 积分公式，

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \omega_i\omega_j f(\xi_i,\eta_j) \tag{2.115}$$

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_i\omega_j\omega_k f(\xi_i,\eta_j,\zeta_k) \tag{2.116}$$

表 2.3　Gauss 积分的求积节点和求积系数

求积节点数 n	求积节点坐标 ξ_i			求积系数 H_i		
1	0.00000	00000	00000	2.00000	00000	00000
2	±0.57735	02691	89626	1.00000	00000	00000
3	±0.77459	66692	41483	0.55555	55555	55556
	0.00000	00000	00000	0.88888	88888	88889
4	±0.86113	63115	94053	0.34785	48451	37454
	±0.33998	10435	84856	0.65214	51548	62546
5	±0.90617	98459	38664	0.23692	68850	56189
	0.53846	93101	05683	0.47862	86704	99366
	0.00000	00000	00000	0.56888	88888	88889
6	±0.93246	95142	03152	0.17132	44923	79170
	±0.66120	93864	66265	0.36076	15730	48139
	±0.23861	91860	83197	0.46791	39345	72691

2.5.2　Hammer 积分

在三角形单元和四面锥单元中，经常会遇到如（2.113）式所示的积分，积分限中包含了

变量自身。Hammer 等[4]导出了有效的积分方案。二维三角形单元及三维四面锥单元的求积节点、求积系数和误差量级分别列于表 2.4 和 2.5 中。

表 2.4 三角形单元的求积节点和求积系数

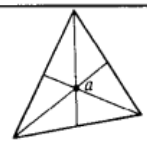
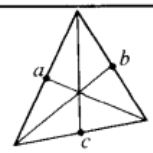
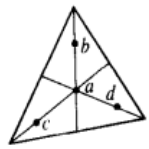
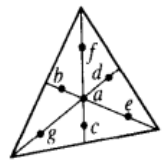
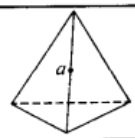
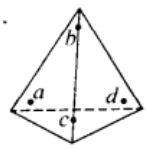
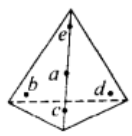
阶次	图形	误差	求积节点	求积节点坐标	求积系数
线性		$R=O(h^2)$	a	1/3,1/3,1/3	1
二次		$R=O(h^3)$	a	1/2,1/2,0	1/3
			b	0,1/2,1/2	1/3
			c	1/2,0,1/2	1/3
三次		$R=O(h^4)$	a	1/3,1/3,1/3	−27/48
			b	0.6, 0.2, 0.2	25/48
			c	0.2, 0.6, 0.2	
			d	0.2, 0.2, 0.6	
四次		$R=O(h^6)$	a	1/3,1/3,1/3	0.22500,00000
			b	α_1,β_1,β_1	0.13239,41527
			c	β_1,α_1,β_1	
			d	β_1,β_1,α_1	
			e	α_2,β_2,β_2	0.12593,91885
			f	β_2,α_2,β_2	
			g	β_2,β_2,α_2	
			其中，	$\alpha_1=0.0597158717\ \ \beta_1=0.4701420641$ $\alpha_2=0.7974269853\ \ \beta_2=0.1012865073$	

表 2.5 四面锥单元的求积节点和求积系数

单元阶次	图形	误差	求积节点	求积节点坐标	求积系数
线性单元		$R=O(h^2)$	a	1/4,1/4,1/4,1/4	1
二次单元		$R=O(h^3)$	a	a,β,β,β	1/4
			b	β,a,β,β	1/4
			c	β,β,α,β	1/4
			d	β,β,β,α	1/4
			$\alpha=0.58541020\ \ \beta=0.13819660$		
三次单元		$R=O(h^4)$	a	1/4 , 1/4 , 1/4 , 1/4	−4/5
			b	1/3,1/6,1/6,1/6	9/20
			c	1/6,1/3,1/6,1/6	9/20
			d	1/6,1/6,1/3,1/6	9/20
			e	1/6,1/6,1/6,1/3	9/20

2.5.3　Labatto 积分

对于弹塑性等材料非线性问题，单元刚度矩阵 $\boldsymbol{K}^e=\int_{V^e}\boldsymbol{B}^T\,\boldsymbol{D}\,\boldsymbol{B}\,\mathrm{d}V$ 中的本构关系矩阵 $\boldsymbol{D}$ 对空间分布来说不再是常数矩阵，因而提高了被积函数的阶次。为保证积分的精度，数值积分的阶次应作相应的提高，一般来说，弹塑性刚度矩阵的积分阶次要比弹性刚度矩阵提高 1~2 阶。

一般情况下，单元体表面的应力最大，有限单元的边界往往首先屈服进入塑性，但是，由于 Gauss 积分方法没有在单元边界上布置积分点，因此不能及时准确地反映单元体进入塑性的时刻。此时，作为一种选择可以考虑采用等间距的 Newton-Cotes 公式。然而，在相同积分点的条件下，利用不等间距内插，且包含边界积分点的 Labatto 积分可以给出更高的求积精度。Labatto 积分公式如下，

$$\int_{-1}^{1}f(\xi)\mathrm{d}\xi=\lambda_1^{(n)}f(-1)+\sum_{k=2}^{n-1}\lambda_k^{(n)}\,f(\xi_k^{(n)})+\lambda_n^{(n)}f(1)\tag{2.117}$$

3~8 阶 Labatto 积分的求积节点和求积系数见表 2.6。

表 2.6　Labatto 积分的求积节点和求积系数

积分阶次 n	求积节点 $\xi_k^{(n)}$	求积系数 $\lambda_k^{(n)}$
3	±1 0	0.33333333 1.33333333
4	±1 ±0.44721360	0.16666667 0.83333333
5	±1 ±0.65465367 0	0.10000000 0.54444444 0.71111111
6	±1 ±0.76505532 ±0.28523152	0.06666667 0.37847496 0.55485838
7	±1 ±0.83022390 ±0.46884879 0	0.04761904 0.27682604 0.43174538 0.48761904
8	±1 ±0.87174015 ±0.59170018 ±0.20929922	0.03571428 0.21070422 0.34112270 0.41245880

2.5.4　数值积分程序

以下给出了两个子程序说明 Gauss 积分的应用。子程序 Gauss 用以获得 Gauss 积分求积节点和求积系数（在附盘 Gauss.f.90 文件中，其他积分方法也在该文件中）。子程序 Elmt 200

以 200 号单元（平面 4~8 节点单元）的刚度矩阵计算为例说明两重 Gauss 积分的应用。

```
      SUBROUTINE Gauss( pgaus, wgaus, ngaus, nerrc )
!.....
!     功能：获取 Gauss 积分的求积节点和求积系数
!     参数：pgaus — 输出量，求积节点数组
!           wgaus — 输出量，求积系数数组
!           ngaus — 输入量，求积节点数
!           nerrc — 输出量，错误标识，非零，表示过程有错、
!.....
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION pgaus( ngaus ), wgaus( ngaus )
      IF( ngaus .EQ. 1 ) THEN
!      一个求积节点
       pgaus( 1 ) =   0.000000000000000D0
       wgaus( 1 ) =   2.000000000000000D0
      ELSE IF( ngaus .EQ. 2 ) THEN
!      两个求积节点
       pgaus( 1 ) = -0.577350269189626D0
       pgaus( 2 ) =   0.577350269189626D0
       wgaus( 1 ) =   1.000000000000000D0
       wgaus( 2 ) =   1.000000000000000D0
      ELSE IF( ngaus .EQ. 3 ) THEN
!      三个求积节点
       pgaus( 1 ) = -0.774596669241483D0
       pgaus( 2 ) =   0.000000000000000D0
       pgaus( 3 ) =   0.774596669241483D0
       wgaus( 1 ) =   0.555555555555556D0
       wgaus( 2 ) =   0.888888888888889D0
       wgaus( 3 ) =   0.555555555555556D0
      ELSE IF( ngaus .EQ. 4 ) THEN
!      四个求积节点
       pgaus( 1 ) = -0.861136311594053D0
       pgaus( 2 ) = -0.339981043584856D0
       pgaus( 3 ) =   0.339981043584856D0
       pgaus( 4 ) =   0.861136311594053D0
       wgaus( 1 ) =   0.347854845137454D0
       wgaus( 2 ) =   0.652145154862546D0
       wgaus( 3 ) =   0.652145154862546D0
       wgaus( 4 ) =   0.347854845137454D0
      ELSE IF( ngaus .EQ. 5 ) THEN
```

```
!         五个求积节点
          pgaus( 1 ) = -0.906179845938664D0
          pgaus( 2 ) = -0.538469310105683D0
          pgaus( 3 ) =  0.000000000000000D0
          pgaus( 4 ) =  0.538469310105683D0
          pgaus( 5 ) =  0.906179845938664D0
          wgaus( 1 ) =  0.236926885056189D0
          wgaus( 2 ) =  0.478628670499366D0
          wgaus( 3 ) =  0.568888888888889D0
          wgaus( 4 ) =  0.478628670499366D0
          wgaus( 5 ) =  0.236926885056189D0
        ELSE IF( ngaus .EQ. 6 ) THEN
!         六个求积节点
          pgaus( 1 ) = -0.932469514203152D0
          pgaus( 2 ) = -0.661209386466265D0
          pgaus( 3 ) = -0.238619186083197D0
          pgaus( 4 ) =  0.238619186083197D0
          pgaus( 5 ) =  0.661209386466265D0
          pgaus( 6 ) =  0.932469514203152D0
          wgaus( 1 ) =  0.171324492379170D0
          wgaus( 2 ) =  0.360761573048139D0
          wgaus( 3 ) =  0.467913934572691D0
          wgaus( 4 ) =  0.467913934572691D0
          wgaus( 5 ) =  0.360761573048139D0
          wgaus( 6 ) =  0.171324492379170D0
        ELSE
          nerrc = 4001
        END IF
        RETURN
        END

        SUBROUTINE Elmt200( ielem, imats, iswth, iuses )
!////////////////////////////////////////////////////////////////////////////////////////////////
!         功能：平面单元
!         参数：主要通过数据模块 ElmtData 传递。
!               ielem   输入量   当前单元序号
!               imats   输入量   当前单元对应的材料类型序号
!               iswth   输入量   调用开关
!               iuses   输入量   当前单元是否是结构组成部分
!////////////////////////////////////////////////////////////////////////////////////////////////
!         在此调用必需的数据模块
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
```

```
!       开设必要的数组
        ......
        DIMENSION pygas( 6 ), wygas( 6 )
        DIMENSION pxgas( 6 ), wxgas( 6 )
!       获得必要的单元信息
        ......
        nxgas = props(   8, imats )      !ξ方向积分点数
        nygas = props(   9, imats )      !η方向积分点数
        IF( nxgas .LE. 0 ) nxgas = 3
        IF( nygas .LE. 0 ) nygas = nxgas
!       获得求积节点和求积系数
        CALL Gauss( pxgas, wxgas, nxgas, nerrc )
        IF( nerrc .GT. 0 ) RETURN
        CALL Gauss( pygas, wygas, nygas, nerrc )
        IF( nerrc .GT. 0 ) RETURN
        ......
!       单元刚度矩阵 stife 零初始化，ndofe 为单元自由度数
        DO idofe = 1, ndofe
          DO jdofe = 1, ndofe
            stife( idofe, jdofe ) = 0.0D0
          END DO
        END DO

        IF( iswth .EQ. 1 ) THEN
!         计算单元刚度矩阵
          kgaus = 0
          DO ixgas = 1, nxgas                          !ξ方向积分
            xloca = pxgas( ixgas )                     !ξ方向积分点
            DO iygas = 1, nygas                        !η方向积分
              kgaus = kgaus + 1
              yloca = pygas( iygas )                   ! η方向积分点
!           计算 B 和 D 矩阵
              CALL Bmat200( shape, bmatx, rcord, ietyp )
              CALL Dmat200( direc, dmatx, elast, poiss, 1, ietyp )
              dvolu = wxgas( ixgas ) * wygas( iygas ) * xjaco * thick
              DO istre = 1, nstre
                DO idofe = 1, ndofe
                  smatx( istre, idofe ) = 0.0D0
                  DO jstre = 1, nstre
                    smatx( istre, idofe ) = smatx( istre, idofe ) +          &
                    dmatx( istre, jstre ) * bmatx( jstre, idofe )
```

```
                END DO
              END DO
            END DO
            DO idofe = 1, ndofe
              DO jdofe = 1, ndofe
                DO istre = 1, nstre
                  stife( idofe, jdofe ) = stife( idofe, jdofe ) +          &
                  bmatx( istre, idofe ) * smatx( istre, jdofe ) *          &
                  dvolu
                END DO
              END DO
            END DO
          END DO
        END DO
      END IF
2000  FORMAT( 20I5 )
2200  FORMAT( /' 致命错误：单元', I5, '为不可压缩材料' /                    &
              '                单元类型 200 只适用可压缩材料!' )
3800  FORMAT( /20x, ' 单　元 ', i5                                        &
            / 7x, 'x', 10x, 'y', 10x,                                     &
                                'S-xx', 10x, 'S-yy', 10x, 'S-zz'          &
            / 5x, 'z(h)', 20x, 'S-yz', 10x, 'S-zx', 10x, 'S-xy ' )
4000  FORMAT( 2x, 2f10.3, 3e14.6 / 2x, f10.3, 10x, 3e14.6 )
      RETURN
      END
```

参考文献

[1]K.J. Bathe, E.L. Wilson. Numerical methods in finite element analysis. Prentice-Hall, Inc, 1976

[2]O.C. Zienkiewicz. The finite element method. Mc Graw-Hill, Inc, 1977

[3]王勖成,邵敏.有限单元法基本原理与数值方法. 北京：清华大学出版社，1988

[4]P.C. Hammer, O.P. Marlows and A.H. Stroud. Numerical integration over simplexes and cones. *Math. Tabls Aids Comp.*, 1956, 10

[5]E. Carrera. A study on arc-length-type methods and their operation failures illustrated by a simple model. *Computers & Structures*, 1994, 50(2).

[6]宋二祥，土工结构安全系数的有限元计算，岩土工程学报，1997，19(2)

[7]李元齐，沈祖炎.弧长控制类方法使用中的若干问题的探讨与改进.计算力学学报，1998，15(4)

[8] P.X.Bellini, A. Chulya. An improved automatic incremental algorithm for the efficient solution of nonlinear finite element equations. *Computers & Structures*, 1987, 26(1/2)

第3章

有限变形分析

对于有限变形问题，由于几何非线性效应，小变形情况下的几何方程和平衡条件不再适用，需要给出有限变形条件下的应变和相应的应力度量以及平衡方程的表述。为了便于读者接受和实用上的方便。本章在笛卡儿坐标系中详细地讨论了有限变形条件下变形的描述和应变的度量。相关内容可以参考专著[1]，更进一步的内容可参考[2-6]。

3.1 物体运动和变形的物质描述和空间描述

在有限变形中，由于物体的变形很大，必须考虑物体受力前后所在空间位置的改变。通常称某一时刻物体在空间占据的区域 V 为物体的构形（Configuration）。称 $t=0$ 时刻物体的构形 V_0 为初始构形，称当前时刻 t 物体的构形为现时构形。

选定一固定的空间直角坐标系，运动物体中每个质点的空间位置可以用一组坐标表示。设初始时刻，$t=0$，某质点的坐标为 $X_i(i=1,2,3)$，并用 X_i 来表示该质点。这个质点不同时刻在空间的位置可用如下方程表示，

$$x_i = x_i(X_j, t) \tag{3.1}$$

如果对于任意的质点 X_j，上述方程都是已知的，则整个物体的运动和变形都已知。如果物体在运动和变形过程中既不互相嵌入，也不分离，则 x_i 是 X_j 的单值连续函数，其雅可比（Jacobi）行列式的值不等于零，即在区域 V_0 内处处有

$$J = \begin{vmatrix} \dfrac{\partial x_1}{\partial X_1} & \dfrac{\partial x_1}{\partial X_2} & \dfrac{\partial x_1}{\partial X_3} \\ \dfrac{\partial x_2}{\partial X_1} & \dfrac{\partial x_2}{\partial X_2} & \dfrac{\partial x_2}{\partial X_3} \\ \dfrac{\partial x_3}{\partial X_1} & \dfrac{\partial x_3}{\partial X_2} & \dfrac{\partial x_3}{\partial X_3} \end{vmatrix} \neq 0 \tag{3.2}$$

在度量物体的运动和变形时需要选取一个特定的构形做为基准，称这个基准构形为参考构形。式（3.1）实际上是以初始构形作为参考构形，用初始时刻质点的坐标 X_i 和时间 t 来描述物体的运动和变形。我们称这种借助于运动着的具体物质点来描述运动和变形的方法叫做物质描述（material description）或拉格朗日（Lagrange）描述，称 X_i 为物质坐标，称 X_i 和 t 为拉格朗日变量。式（3.1）实质上给出的是初始位置坐标为 X_i 的质点的运动轨迹，质点运动

的速度和加速度分别是

$$v_i(X_j,t)=\frac{\partial}{\partial t}x_i(X_j,t) \tag{3.3}$$

$$a_i(X_j,t)=\frac{\partial^2}{\partial t^2}x_i(X_j,t) \tag{3.4}$$

同样，我们也可以选用现时构形 V 为参考构形，用现时构形中质点的坐标 x_i 和时间 t 来描述物体的运动和变形，

$$X_i=X_i(x_j,t) \tag{3.5}$$

我们称这种描述方法为空间描述（spatial description）或欧拉（Euler）描述，称 x_i 为空间坐标，x_i 和 t 为欧拉变量。

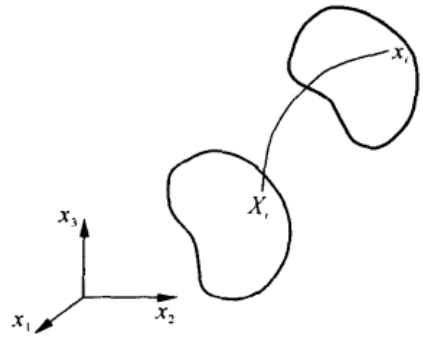


图 3.1　物体运动描述

空间描述的研究方式不是跟随运动的质点研究其运动状态，而是研究不同时刻经过同一空间点 x_i 的质点的运动状态。

在空间描述中物体的瞬时运动常用速度矢量场来描述。

$$v_i=v_i(x_j,t) \tag{3.6}$$

式中，v_i 为 t 时刻占据 x_j 坐标的质点的运动速度。

值得注意的是，位于 x_i 的空间点和位于这个空间点的物质质点是有区别的。$v_i(x_j,t)$ 和 $v_i(x_j,t+\mathrm{d}t)$ 分别代表同一空间点不同时刻观测到的速度，但不一定是同一个质点在不同时刻的速度。因此 $\dfrac{\partial v_i}{\partial t}$ 不一定代表 t 时刻占据空间位置 x_j 的质点的加速度。因为时刻 t 位于 x_j 的那个质点在 $t+\mathrm{d}t$ 时刻已运动到 $x_j+v_j\mathrm{d}t$ 处，因此 t 时刻位于 x_j 的质点的加速度 a_i 应当满足

$$a_i(x_j,t)\mathrm{d}t=v_i(x_j+v_j\mathrm{d}t,t+\mathrm{d}t)-v_i(x_j,t)$$

所以，

$$a_i=\frac{\partial v_i}{\partial t}+v_j\frac{\partial v_i}{\partial x_j} \tag{3.7}$$

式中，相同角标满足爱因斯坦求和约定。如果没有特别说明，本章其他公式也沿用这一约定。式（3.7）定义的加速度是与运动着的物质质点相关联的，因此称为速度的物质导数。式（3.7）中右端第一项可以解释为由速度场的时间相关性引起的，称为加速度的当地部分；第二项是由非均匀速度场中质点运动的贡献，称为加速度的对流部分。任何与运动着的质点相关联的物理量 F（标量、矢量或张量）都可以类似地定义物质导数，

$$\dot{F}=\frac{\partial F}{\partial t}+v_j\frac{\partial F}{\partial x_j} \tag{3.8}$$

3.2　变形梯度、体元、面元

本节以初始构形作为参考构形，即采用物质描述方法，分析线元、面元和体元在有限变形过程中的变化。

3.2.1 变形梯度

在参考构形中，两个相邻质点的坐标为 X_i 和 $X_i+\mathrm{d}X_i$，在时刻 t 这两个质点在现时构形中的坐标分别为 x_i 和 $x_i+\mathrm{d}x_i$（如图 3.2 所示），两点间的距离为

$$\mathrm{d}x_i = x_i(X_j+\mathrm{d}X_j,t)-x_i(X_j,t)$$

对 $x_i(X_j+\mathrm{d}X_j,t)$ 在 (X_j,t) 点作 Taylor 展开，略去高阶项，可得

$$\mathrm{d}x_i = \frac{\partial x_i}{\partial X_j}\mathrm{d}X_j \tag{3.9}$$

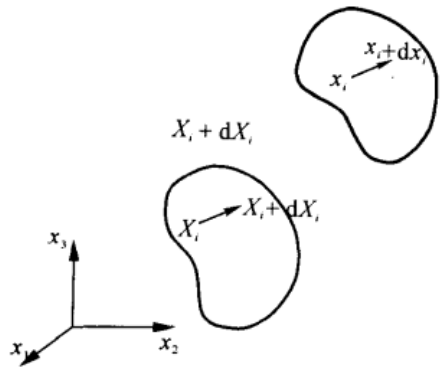


图 3.2 线元示意图

称物体的现时构形坐标 x_i 对于物质坐标 X_j 的偏导数 $\dfrac{\partial x_i}{\partial X_j}$ 为变形梯度。可以证明，变形梯度是一个非对称的二阶张量。从式（3.9）可以看出，变形梯度相当于一种线性变换，它把参考构形中质点 X_i 的邻域映射到现时构形中空间点 x_i 的一个邻域。或者说，它把初始的线元 $\mathrm{d}X_i$ 变换到现时的线元 $\mathrm{d}x_i$。因此变形梯度张量刻画了整个变形，既包括线元的伸缩，也包括了线元的转动。

比较式（3.2）可以看出，J 正是变形梯度矩阵的行列式。为简便起见，定义排序张量 e_{ijk}，当下标 i,j,k 是按 1，2，3 次序偶排列时 e_{ijk} 取值为 1；奇排列时取值为–1；其余情形为 0。具体地说，

$$e_{123}=e_{231}=e_{312}=1$$

$$e_{213}=e_{132}=e_{321}=-1$$

$$e_{111}=e_{222}=e_{132}=e_{112}=e_{113}=\cdots=0$$

利用排序张量 e_{ijk}，式（3.2）可以表示为

$$J=e_{ijk}\frac{\partial x_i}{\partial X_1}\frac{\partial x_j}{\partial X_2}\frac{\partial x_k}{\partial X_3} \tag{3.10}$$

3.2.2 体元

现在我们先来考虑初始构形中由 $QQ'Q''Q'''$ 四个相邻物质点构成的四面体体元，它的三个棱线元 QQ'、QQ'' 和 QQ''' 的坐标分量分别为 $\mathrm{d}X_p$，δX_l 和 ΔX_m。在现时构形中对应的四面体为 $qq'q''q'''$，对应的线元 qq'，qq'' 和 qq''' 的坐标分量分别记为 $\mathrm{d}x_i,\delta x_j,\Delta x_k$，如图 3.3 所示。根据式（3.9）有

$$\mathrm{d}x_i=\frac{\partial x_i}{\partial X_p}\mathrm{d}X_p \qquad \delta x_j=\frac{\partial x_i}{\partial X_l}\delta X_l \qquad \Delta x_k=\frac{\partial x_k}{\partial x_m}\Delta X_m \tag{3.11}$$

现时构形中体元的体积为

$$dV = (d\boldsymbol{x} \times \delta\boldsymbol{x}) \cdot \Delta\boldsymbol{x} / 6$$

将上式写成矩阵形式，并用排序张量展开，得

$$dV = \begin{vmatrix} dx_1 & dx_2 & dx_3 \\ \delta x_1 & \delta x_2 & \delta x_3 \\ \Delta x_1 & \Delta x_2 & \Delta x_3 \end{vmatrix} / 6 = e_{ijk} dx_i \delta x_j \Delta x_k / 6$$

将(3.11)式代入上式，得

$$dV = e_{ijk} \frac{\partial x_i}{\partial X_p} \frac{\partial x_j}{\partial X_l} \frac{\partial x_k}{\partial X_m} dX_p \delta X_l \Delta X_m / 6$$

利用排序张量的定义和式(3.10)式，上式可化简为

$$dV = e_{plm} J dX_p \delta X_l \Delta X_m / 6 \tag{3.12}$$

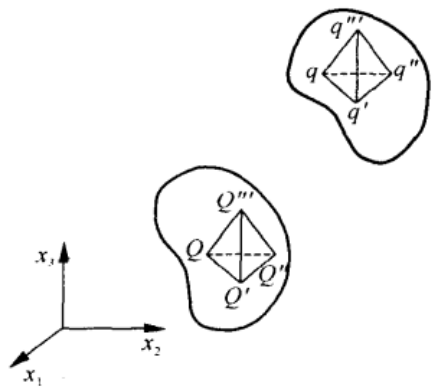


图 3.3　体元示意图

同理可得到初始构形中微元体的体积为

$$dV_0 = (d\boldsymbol{X} \times \delta\boldsymbol{X}) \cdot \Delta\boldsymbol{X} / 6 = e_{plm} dX_p \delta X_l \Delta X_m / 6 \tag{3.13}$$

比较(3.12)和(3.13)式可见，现时构形中的体元体积与相应初始构形中体元的体积满足如下关系，

$$dV = J dV_0 \tag{3.14}$$

如果用 ρ_0 和 ρ 分别表示初始构形和现时构形中体元内介质的密度，那么由质量守恒 $\rho_0 dV_0 = \rho dV$，可得

$$J\rho = \rho_0 \tag{3.15}$$

式（3.14）表明，J 代表了体元在变形期间体积的变化率。当介质为不可压缩材料时，即变形期间体元的密度和体积没有变化，则有

$$J = 1 \tag{3.16}$$

3.2.3　面元

下面我们再来考虑物体变形期间面元的大小和方位的变化。初始构形中由 $QQ'Q''$ 组成的三角面元在现时构形中为 $qq'q''$。在初始构形中，线元 QQ' 和 QQ'' 的坐标分量分别为 dX_i 和 δX_i。对应于现时构形中线元 qq' 和 qq''，其坐标分量分别为 dX_i 和 δX_i。分别用 $d\boldsymbol{A}_0$、$\boldsymbol{N}_i$ 和 dA_0 代表初始构形中三角形面元的有向面积、单位法向矢量和面积的大小；用 $d\boldsymbol{A}$、$\boldsymbol{n}_i$ 和 dA 表示该面元在现时构形中的有向面积、单位法向矢量和面积大小，如图 3.4 所示。则

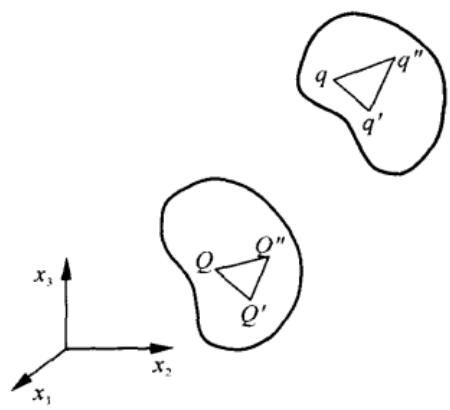


图 3.4　面元示意图

$$d\boldsymbol{A}_0 = dA_0 \cdot \boldsymbol{N} = d\boldsymbol{X} \times \delta\boldsymbol{X} / 2$$

$$d\boldsymbol{A} = dA\,\boldsymbol{n} = d\boldsymbol{x} \times \delta\boldsymbol{x} / 2$$

注意到

$$d\boldsymbol{X} \times \delta\boldsymbol{X} = e_{ijk} dX_j \delta X_k \boldsymbol{i}_i$$

$$\mathrm{d}\boldsymbol{x}\times\delta\boldsymbol{x}=e_{ijk}\mathrm{d}x_j\delta x_k\boldsymbol{i}_i$$

式中，$\boldsymbol{i}_i$ 为沿第 i 个坐标轴正向的单位矢量。由上面两式不难得到

$$N_i\mathrm{d}A_0=e_{ijk}\mathrm{d}X_j\delta X_k/2 \tag{3.17}$$

$$n_i\mathrm{d}A=e_{ijk}\mathrm{d}x_j\delta x_k/2 \tag{3.18}$$

将（3.18）式的两端分别乘以 $\partial x_i/\partial X_p$，利用（3.9）式可得

$$\frac{\partial x_i}{\partial X_p}n_i\mathrm{d}A=\frac{1}{2}e_{ijk}\frac{\partial x_i}{\partial X_p}\frac{\partial x_j}{\partial X_l}\frac{\partial x_k}{\partial X_m}\mathrm{d}X_l\delta X_m=\frac{1}{2}Je_{plm}\mathrm{d}X_l\delta X_m$$

将上式与(3.17)式比较，得

$$\frac{\partial x_i}{\partial X_p}n_i\mathrm{d}A=JN_p\mathrm{d}A_0 \tag{3.19}$$

注意到（3.19）式中 p 为自由指标，取值 1,2,3，因此它可以看成为以 $n_i\mathrm{d}A$ 为未知量的三阶线性方程组，方程组的系数矩阵即为变形梯度矩阵。由于 $J\neq0$，因此变形梯度矩阵有逆。注意到

$$\frac{\partial x_i}{\partial X_k}\frac{\partial X_k}{\partial x_j}=\delta_{ij} \tag{3.20}$$

式中，δ_{ij} 为克朗内克记号。因此变形梯度矩阵的逆矩阵为 $\partial X_i/\partial x_j$，（3.19）式因此也可以表示为

$$n_i\mathrm{d}A=J\frac{\partial X_p}{\partial x_i}N_p\mathrm{d}A_0 \tag{3.21}$$

根据克莱姆法则，$\partial X_i/\partial x_j$ 可用变形梯度张量的分量表示，即

$$\frac{\partial X_i}{\partial x_j}=\frac{1}{2J}e_{imk}e_{jnl}\frac{\partial x_k}{\partial X_m}\frac{\partial x_l}{\partial X_n} \tag{3.22}$$

3.3 格林应变、阿耳曼西应变和柯西应变

本节讨论有限变形条件下，变形体的应变度量。

3.3.1 格林(Green)应变

物体由初始状态变形到现时状态时，用变形梯度可以刻画物体的变形，但有时人们更关心 Q 点附近的相对变形，即应变。如图 3.4 所示，初始构形中的线元 QQ' 和 QQ''，变形到现时构形中线元 qq' 和 qq''，两者的差别可以用线元矢量点积的差来表示，即

$$\mathrm{d}x_i\delta x_i-\mathrm{d}X_i\delta X_i=\left(\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j}-\delta_{ij}\right)\mathrm{d}X_i\delta X_j \tag{3.23}$$

可以将此式等号右端的因子作为应变的度量，定义格林应变为

$$E_{ij}=\frac{1}{2}\left(\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j}-\delta_{ij}\right) \tag{3.24}$$

格林应变定义中含有系数 1/2，目的是在小应变情况下，格林应变可以退化到柯西应变。注意到，格林应变是一个无量纲量，具有九个分量。可以证明，格林应变是一个二阶对称张量，因此只有六个独立分量。

显然，如果点 Q 的邻域内变形过程仅为刚体运动，则初始构形和现时构形中的三角形具有相同的形状，（3.23）式的右端恒为零。由于 $\mathrm{d}\boldsymbol{X}$ 和 $\delta\boldsymbol{X}$ 是任意的，因此（3.23）式右端的因子在 Q 邻域作刚体运动时恒为零，即格林应变的所有分量均为零。

利用格林应变 E_{ij}，(3.23)式可表示为

$$\mathrm{d}x_i\delta x_i-\mathrm{d}X_i\delta X_i=2E_{ij}\mathrm{d}X_i\delta X_j \tag{3.25}$$

注意到 $\mathrm{d}X_i$ 和 δX_i 是任取的。为说明 E_{ij} 的物理意义，令 QQ' 和 QQ'' 分别平行于坐标轴 x_m 和 x_n，即

$$\mathrm{d}X_i=\mathrm{d}S\delta_{mi}\qquad \delta X_j=\delta S\delta_{nj}$$

式中，$\mathrm{d}S$ 和 δS 分别为线元 $\mathrm{d}\boldsymbol{X}$ 和 $\delta\boldsymbol{X}$ 的长度。

将上式代入（3.25）式，可得

$$2\mathrm{d}S\,\delta S\,E_{mn}=(\mathrm{d}x_i\delta x_i-\delta_{mn}\mathrm{d}S\,\delta S)$$

设矢量 $\mathrm{d}\boldsymbol{x}$，$\delta\boldsymbol{x}$ 的长度分别为 $\mathrm{d}s$ 和 δs，夹角为 θ，则

$$2E_{mn}=\frac{\mathrm{d}s\,\delta s}{\mathrm{d}S\,\delta S}\cos\theta-\delta_{mn}$$

当 $m=n$，且 $\mathrm{d}S=\delta S$（即 Q' 和 Q'' 为同一点）时，q' 和 q'' 重合，$\mathrm{d}s=\delta s$，$\theta=0$，$\mathrm{d}x_i\,\delta x_i=(\mathrm{d}s)^2$。则有

$$E_{\overline{mm}}=\frac{1}{2}\left[\left(\frac{\mathrm{d}s}{\mathrm{d}S}\right)^2-1\right] \tag{3.26}$$

上式代表沿 x_m 轴方向线元长度的变化。注意，上式中下标 m 上加顶标表示不求和，以下同。

当 $m\neq n$ 时，

$$E_{mn}=\frac{1}{2}\frac{\mathrm{d}s\,\delta s}{\mathrm{d}S\,\delta S}\cos\theta$$

用 α 代表变形前后线元间夹角的改变，即

$$\alpha=\frac{\pi}{2}-\theta$$

则有

$$E_{mn}=\frac{1}{2}\frac{\mathrm{d}s}{\mathrm{d}S}\frac{\delta s}{\delta S}\sin\alpha \tag{3.27}$$

由上式可以看出，当 $m\neq n$ 时，E_{mn} 能反映沿 x_m 和 x_n 轴方向两个线元间夹角的变化。但其力学解释不如小变形情况下那么简单，因为 E_{mn} 还跟线元长度变化有关。

3.3.2 阿耳曼西(Almansi)应变

格林应变是以初始构形为参考构形定义的应变，若以现时构形作为参考构形，则与（3.23）式类似，有

$$\mathrm{d}x_i\,\delta x_i-\mathrm{d}X_i\,\delta X_i=\left(\delta_{ij}-\frac{\partial X_k}{\partial x_i}\frac{\partial X_k}{\partial x_j}\right)\mathrm{d}x_i\,\delta x_j \tag{3.28}$$

可以类似地定义阿耳曼西（Almansi）应变张量为

$$e_{ij}=\frac{1}{2}\left(\delta_{ij}-\frac{\partial X_k}{\partial x_i}\frac{\partial X_k}{\partial x_j}\right) \tag{3.29}$$

阿耳曼西应变张量各分量的物理意义与格林应变类似，读者可以自己试着推导。同样，阿耳曼西应变张量也是对称张量。

3.3.3 位移表示的应变

通常，现时构形和初始构形的差别可以用位移矢量u_i来表示。若以初始构形作为参考构形，则位移矢量用物质坐标表示为

$$u_i=x_i(X_j,t)-X_i \tag{3.30}$$

若以现时构形作为参考构形，则位移矢量可定义为

$$u_i=x_i-X_i(x_j,t) \tag{3.31}$$

相应的变形梯度张量可以用位移分量表示为

$$\frac{\partial x_i}{\partial X_j}=\frac{\partial u_i}{\partial X_j}+\delta_{ij} \tag{3.32}$$

$$\frac{\partial X_i}{\partial x_j}=\delta_{ij}-\frac{\partial u_i}{\partial x_j} \tag{3.33}$$

称$\frac{\partial u_i}{\partial X_j}$和$\frac{\partial u_i}{\partial x_j}$为相对于初始构形和现时构形的位移梯度张量。分别把（3.32）和（3.33）式代入（3.24）和（3.29）式，可得位移表示的格林应变和阿耳曼西应变，

$$E_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial X_j}+\frac{\partial u_j}{\partial X_i}+\frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}\right) \tag{3.34}$$

$$e_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right) \tag{3.35}$$

要注意的是，在计算格林应变时，位移分量被看作是物理坐标X_i的函数，而在计算阿耳曼西应变时，u_i被看作是空间坐标的函数。

由(3.34)和(3.35)式可以看出，格林应变和阿耳曼西应变均有两部分组成：线性部分和非线性部分。为方便起见，定义

$$E_{ij}^0 = E_{ji}^0 = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right)$$
$$R_{ij}^0 = -R_{ji}^0 = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} - \frac{\partial u_j}{\partial X_i}\right) \tag{3.36}$$

$$e_{ij}^0 = e_{ji}^0 = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$
$$r_{ij}^0 = -r_{ji}^0 = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{3.37}$$

式中，E_{ij}^0 和 e_{ij}^0 分别为 E_{ij} 和 e_{ij} 的线性部分。可以证明，R_{ij}^0 和 r_{ij}^0 分别代表物质描述和空间描述时微元体的刚体转动。由(3.36)和(3.37)式可得

$$\frac{\partial u_i}{\partial X_j} = E_{ij}^0 + R_{ij}^0 \qquad \frac{\partial u_i}{\partial x_j} = e_{ij}^0 + r_{ij}^0 \tag{3.38}$$

从而得到 E_{ij} 和 e_{ij} 的另一种表达形式，

$$E_{ij} = E_{ij}^0 + \frac{1}{2}(E_{ki}^0 + R_{ki}^0)(E_{kj}^0 + R_{kj}^0) \tag{3.39}$$

$$e_{ij} = e_{ij}^0 - \frac{1}{2}(e_{ki}^0 + r_{ki}^0)(e_{kj}^0 + r_{kj}^0) \tag{3.40}$$

3.3.4　柯西(Cauchy)应变

在小变形情况下，即应变和转动都很小，此时位移的梯度张量与单位 1 相比很小，

$$\frac{\partial u_i}{\partial X_j} << 1, \quad \frac{\partial u_i}{\partial x_j} << 1 \tag{3.41}$$

可以证明

$$\frac{\partial}{\partial x_i} \approx \frac{\partial}{\partial X_i} \tag{3.42}$$

上式意味着，在微小变形条件下，对坐标的微商运算不需要区别质点在现时构形中的坐标 x_i 和在初始构形中的坐标 X_i。此时（3.34）或（3.35）式中非线性乘积项相对线性项很小，可以忽略不计。格林应变和阿耳曼西应变退化为柯西应变 ε_{ij}，

$$\varepsilon_{ij} = E_{ij} = e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) \tag{3.43}$$

当应变很小，但转动较大，且 E_{ij}^0 和 $(R_{ij}^0)^2$ 同量级，或 e_{ij}^0 和 $(r_{ij}^0)^2$ 同量级，此时，（3.34）和（3.35）式可简化为

$$E_{ij} = E_{ij}^0 + \frac{1}{2}R_{ki}^0 R_{kj}^0 \qquad e_{ij} = e_{ij}^0 - \frac{1}{2}r_{ki}^0 r_{kj}^0 \tag{3.44}$$

一些板壳的大挠度问题往往属于此类，且一般情况下 R_{ij}^0 或 r_{ij}^0 中只有个别分量较大，因此问题还可以进一步简化。

可以证明，在物体内的任意点至少有三个相互垂直的应变主轴，对应主轴坐标系的应变张量非对角分量（当$i \neq j$时）全为零，即平行于主轴且互相正交的物质线元变形后仍保持正交状态。

例 3-1 一物体变形前后满足如下规律，

$$\begin{cases} x_1 = X_1 \cos\Omega - X_2 \sin\Omega \\ x_2 = X_1 \sin\Omega + X_2 \cos\Omega \\ x_3 = X_3 \end{cases} \tag{3.45}$$

现分析物体中任意点的应变。

根据格林应变的定义(3.24)式，可直接求得

$$E_{11} = \frac{1}{2}\left(\frac{\partial x_1}{\partial X_1}\frac{\partial x_1}{\partial X_1} + \frac{\partial x_2}{\partial X_1}\frac{\partial x_2}{\partial X_1} + \frac{\partial x_3}{\partial X_1}\frac{\partial x_3}{\partial X_1} - 1\right) = 0$$

同样可以求得$E_{22} = E_{33} = 0$，$E_{23} = E_{31} = E_{12} = 0$。

由(3.45)式容易求得物质坐标描述的位移分量为

$$\begin{cases} u_1 = X_1(\cos\Omega - 1) - X_2 \sin\Omega \\ u_2 = X_1 \sin\Omega + X_2(\cos\Omega - 1) \\ u_3 = 0 \end{cases}$$

按(3.34)式同样可以直接验证格林应变分量都等于零。

为求阿耳曼西应变张量，对(3.45)式作适当的变换，将物质坐标用空间坐标表示，

$$\begin{cases} X_1 = x_1 \cos\Omega + x_2 \sin\Omega \\ X_2 = -x_1 \sin\Omega + x_2 \cos\Omega \\ X_3 = x_3 \end{cases} \tag{3.46}$$

同样可求得用空间坐标描述的位移分量为

$$\begin{cases} u_1 = x_1(1 - \cos\Omega) - x_2 \sin\Omega \\ u_2 = x_1 \sin\Omega + x_2(1 - \cos\Omega) \\ u_3 = 0 \end{cases}$$

根据阿耳曼西应变张量的定义，可直接求得

$$e_{11} = \frac{1}{2}\left(1 - \frac{\partial X_1}{\partial x_1}\frac{\partial X_1}{\partial x_1} - \frac{\partial X_2}{\partial x_1}\frac{\partial X_2}{\partial x_1} - \frac{\partial X_3}{\partial x_1}\frac{\partial X_3}{\partial x_1}\right) = 0$$

同样可以求得$e_{22} = e_{33} = 0$，$e_{23} = e_{31} = e_{12} = 0$。

如果忽略非线性项的影响，则有

$$E_{11}^0 = E_{22}^0 = \cos\Omega - 1 \qquad E_{33}^0 = 0 \qquad E_{23}^0 = E_{31}^0 = E_{12}^0 = 0$$

$$e_{11}^0 = e_{22}^0 = 1 - \cos\Omega \qquad e_{33}^0 = 0 \qquad e_{23}^0 = e_{31}^0 = e_{12}^0 = 0$$

不难看出，式(3.45)表示物体绕$X_3(x_3)$作刚体旋转，因此应变分量应为零，格林应变和阿耳曼西应变均反映了这一特性；而定义在初始构形和现时构形中的柯西应变却产生了正应变，不符合客观实际。

3.4　变形率

物质描述法在固体力学分析中经常用到，但这种描述方法并不总是方便的。如在描述轧制过程中钢材的变形时，我们没有必要知道钢材中每个质点的整个运动的历史，只需要描绘出各个时刻位于指定的空间各点处的那些质点的运动，而不管这些质点来自何处。即我们感兴趣的是瞬时速度场和它们随时间的演化，这时采用空间描述更加方便。

根据物质导数的概念，质量密度 $\rho(x_i,t)$ 的物质导数是

$$\dot{\rho}=\frac{\partial\rho}{\partial t}+v_j\frac{\partial\rho}{\partial x_j} \tag{3.47}$$

如果在空间固定的坐标系内考虑一个无限小的空间体积元，可以认为流入这个体元的质量的净速率等于质量积累的速率，于是我们有连续性方程

$$\frac{\partial}{\partial x_1}(\rho v_1)+\frac{\partial}{\partial x_2}(\rho v_2)+\frac{\partial}{\partial x_3}(\rho v_3)+\frac{\partial\rho}{\partial t}=0 \tag{3.48}$$

利用质量密度的物质导数，上式可以表示为

$$\dot{\rho}+\rho\frac{\partial v_i}{\partial x_i}=0 \tag{3.49}$$

连续性方程在物理上表示为质量守恒。如果密度的变化是非常小的，以致可以忽略不计，则连续性方程（3.48）可以简化为

$$\frac{\partial v_i}{\partial x_i}=\frac{\partial v_1}{\partial x_1}+\frac{\partial v_2}{\partial x_2}+\frac{\partial v_3}{\partial x_3}=0 \tag{3.50}$$

考虑一个典型的质点 P，它在时刻 t 的瞬时坐标为的 x_i，瞬时速度为 $v_i(x_j,t)$；P' 为 P 的邻域内的另外一个质点，t 时刻的瞬时坐标为 $x_i+\mathrm{d}x_i$，速度为 $v_i(x_j+\mathrm{d}x_j,t)$。则 t 时刻 P' 相对于 P 的运动速度为

$$\mathrm{d}v_i=\frac{\partial v_i}{\partial x_j}\mathrm{d}x_j \tag{3.51}$$

称 $\partial v_i/\partial x_j$ 为速度梯度，它表达了速度场在空间的变化规律。与变形梯度类似，速度梯度是一个非对称张量，它可以分解为对称和反对称两部分之和，即

$$\frac{\partial v_i}{\partial x_j}=\frac{1}{2}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)+\frac{1}{2}\left(\frac{\partial v_i}{\partial x_j}-\frac{\partial v_j}{\partial x_i}\right) \tag{3.52}$$

称反对称部分为旋率张量 Ω_{ij}，对称部分为变形率张量 V_{ij}，即

$$\Omega_{ij}=-\frac{1}{2}\left(\frac{\partial v_i}{\partial x_j}-\frac{\partial v_j}{\partial x_i}\right) \tag{3.53}$$

$$V_{ij}=\frac{1}{2}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right) \tag{3.54}$$

因此，速度梯度可以表示为

$$\frac{\partial v_i}{\partial x_j} = -\Omega_{ij} + V_{ij} \tag{3.55}$$

把（3.55）式代入（3.51）式，并定义

$$\mathrm{d}v_i^{\bullet} = -\Omega_{ij}\mathrm{d}x_j \qquad \mathrm{d}v_i^{\bullet\bullet} = V_{ij}\mathrm{d}x_j \tag{3.56}$$

则

$$\mathrm{d}v_i = \mathrm{d}v_i^{\bullet} + \mathrm{d}v_i^{\bullet\bullet} \tag{3.57}$$

由于 Ω_{ij} 是反对称的，因此总存在一个对偶矢量 Ω_k，使得

$$\Omega_k = \frac{1}{2}e_{kij}\Omega_{ij} \tag{3.58}$$

称 Ω_k 为旋度矢量。上式两端同乘以 e_{klm}，利用恒等式

$$e_{kij}e_{klm} = \delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}$$

可得

$$\Omega_{ij} = e_{ijk}\Omega_k \tag{3.59}$$

把（3.59）式代入（3.56）得

$$\mathrm{d}v_i^{\bullet} = -e_{ijk}\Omega_k\mathrm{d}x_j = (\boldsymbol{\Omega}\times\mathrm{d}\boldsymbol{x})_i \tag{3.60}$$

上式表明，相对速度 $\mathrm{d}\boldsymbol{v}^{\bullet}$ 为旋度矢量 $\boldsymbol{\Omega}$ 和 $\mathrm{d}\boldsymbol{x}$ 的矢量积，其物理意义是质点 P 的邻域绕过 P 点的某个轴的刚体转动，矢量 Ω_k 为转动角速度。因此，如果质点 P 的速度梯度本身是反对称的，即其对称部分为零，那么 P 点邻域瞬时地做刚体运动。反过来，如果质点 P 的邻域的速度 v_i 只有绕 P 点的转动速度部分，容易证明此时速度梯度为反对称张量。

类似于应变的概念，我们来考虑相对速度的瞬时变化率。在 P 点（坐标为 x_i）的邻域内任取两点 P'（坐标为 $x_i+\mathrm{d}x_i$）和 P''（坐标为 $x_i+\delta x_i$），考察线元 $\mathrm{d}\boldsymbol{x}$ 和 $\delta\boldsymbol{x}$ 点积的变化率，

$$(\mathrm{d}x_i\ \delta x_i)^{\bullet} = \mathrm{d}v_i\ \delta x_i + \mathrm{d}x_i\ \delta v_i$$

将(3.51)式代入上式可得

$$(\mathrm{d}x_i\ \delta x_i)^{\bullet} = \frac{\partial v_i}{\partial x_j}\mathrm{d}x_j\ \delta x_i + \frac{\partial v_i}{\partial x_j}\mathrm{d}x_i\ \delta x_j$$

注意到上式等号右边第二项中的下标 i 和 j 均为求和指标，互易不影响结果，因此

$$(\mathrm{d}x_i\ \delta x_i)^{\bullet} = \frac{\partial v_i}{\partial x_j}\mathrm{d}x_j\ \delta x_i + \frac{\partial v_j}{\partial x_i}\mathrm{d}x_j\ \delta x_i = \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\mathrm{d}x_j\ \delta x_i$$

利用(3.54)式可得

$$(\mathrm{d}x_i\ \delta x_i)^{\bullet} = 2V_{ij}\mathrm{d}x_i\ \delta x_j \tag{3.61}$$

若物体作刚体运动，运动前后的三角形保持不变，即 $(\mathrm{d}x_i\ \delta x_i)^{\bullet} = 0$，注意到 $\mathrm{d}x_i$ 和 δx_j 的任意性，当物体作刚体运动时，有 $V_{ij} \equiv 0$。

下面我们来分析变形率张量和应变变化率的关系，对格林应变关于时间求物质导数，

$$\dot{E}_{ij} = \frac{1}{2}\left(\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} - \delta_{ij}\right)^{\bullet} = \frac{1}{2}\left(\frac{\partial v_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} + \frac{\partial x_k}{\partial X_i}\frac{\partial v_k}{\partial X_j}\right)$$

根据复合函数的求导原则，

$$\frac{\partial}{\partial X_i}=\frac{\partial}{\partial x_l}\frac{\partial x_l}{\partial X_i}$$

上式可改写为

$$\dot{E}_{ij}=\frac{1}{2}\left(\frac{\partial v_k}{\partial x_l}\frac{\partial x_l}{\partial X_i}\frac{\partial x_k}{\partial X_j}+\frac{\partial v_k}{\partial x_l}\frac{\partial x_l}{\partial X_j}\frac{\partial x_k}{\partial x_i}\right)$$

互易上式等号右边第二项中的求和指标 k 和 l，得

$$\dot{E}_{ij}=\frac{1}{2}\left(\frac{\partial v_k}{\partial x_l}\frac{\partial x_l}{\partial X_i}\frac{\partial x_k}{\partial X_j}+\frac{\partial v_l}{\partial x_k}\frac{\partial x_k}{\partial X_j}\frac{\partial x_l}{\partial x_i}\right)=\frac{1}{2}\left(\frac{\partial v_k}{\partial x_l}+\frac{\partial v_l}{\partial x_k}\right)\frac{\partial x_l}{\partial X_i}\frac{\partial x_k}{\partial X_j}$$

利用(3.54)式可得

$$\dot{E}_{ij}=\frac{\partial x_k}{\partial X_i}\frac{\partial x_l}{\partial X_j}V_{kl} \tag{3.62}$$

对上式求逆，

$$V_{ij}=\frac{\partial X_k}{\partial x_i}\frac{\partial X_l}{\partial x_j}\dot{E}_{kl} \tag{3.63}$$

同样，对 Almansi 应变求物质导数，

$$\dot{e}_{ij}=\frac{1}{2}\left(\delta_{ij}-\frac{\partial X_k}{\partial x_i}\frac{\partial X_k}{\partial x_j}\right)^{\cdot}=-\frac{1}{2}\left(\frac{\partial X_k}{\partial x_i}\right)^{\cdot}\frac{\partial X_k}{\partial x_j}-\frac{1}{2}\frac{\partial X_k}{\partial x_i}\left(\frac{\partial X_k}{\partial x_i}\right)^{\cdot} \tag{3.64}$$

考虑到固定质点的物质坐标 X_i 是不随时间改变而改变的，因此物质坐标 X_i 的物质导数为零，即

$$\dot{X}_i=\frac{\partial X_i}{\partial t}+v_j\frac{\partial X_i}{\partial x_j}=0$$

对上式关于 x_k 求偏导数，则

$$\frac{\partial^2 X_i}{\partial t\partial x_k}+\frac{\partial v_j}{\partial x_k}\frac{\partial X_i}{\partial x_j}+v_j\frac{\partial^2 X_i}{\partial x_j\partial x_k}=0$$

注意到

$$\frac{\partial^2 X_i}{\partial t\partial x_k}+v_j\frac{\partial^2 X_i}{\partial x_j\partial x_k}=\frac{\partial}{\partial t}\left(\frac{\partial X_i}{\partial x_k}\right)+v_j\frac{\partial}{\partial x_j}\left(\frac{\partial X_i}{\partial x_k}\right)=\left(\frac{\partial X_i}{\partial x_k}\right)^{\cdot}$$

则

$$\left(\frac{\partial X_i}{\partial x_k}\right)^{\cdot}=-\frac{\partial v_j}{\partial x_k}\frac{\partial X_i}{\partial x_j} \tag{3.65}$$

把（3.65）式代入（3.64）式可得

$$\dot{e}_{ij}=\frac{1}{2}\frac{\partial v_l}{\partial x_i}\frac{\partial X_k}{\partial x_l}\frac{\partial X_k}{\partial x_j}+\frac{1}{2}\frac{\partial X_k}{\partial x_i}\frac{\partial v_l}{\partial x_j}\frac{\partial X_k}{\partial x_l}$$

利用 Almansi 应变的定义可得

$$\dot{e}_{ij}=\frac{1}{2}\frac{\partial v_l}{\partial x_i}(\delta_{lj}-2e_{lj})+\frac{1}{2}\frac{\partial v_l}{\partial x_j}(\delta_{il}-2e_{il})=\frac{1}{2}\left(\frac{\partial v_j}{\partial x_i}+\frac{\partial v_i}{\partial x_j}\right)-\left(\frac{\partial v_l}{\partial x_i}e_{lj}+\frac{\partial v_l}{\partial x_j}e_{il}\right)$$

利用(3.65)式，上式可简化为

$$\dot{e}_{ij}=V_{ij}-e_{lj}V_{li}-e_{li}V_{lj}+e_{lj}\Omega_{li}+e_{li}\Omega_{lj} \tag{3.66}$$

由此可见，Almansi 应变张量的物质变化率与刚体转动有关。为方便描述物质的应力-应变关系起见，定义 Almansi 应变的本构速度，

$$\overset{\nabla}{e}_{ij}=\dot{e}_{ij}-e_{lj}\Omega_{li}-e_{li}\Omega_{lj} \tag{3.67}$$

它与变形率张量的关系为

$$\overset{\nabla}{e}_{ij}=V_{ij}-e_{il}V_{lj}-e_{jl}V_{li} \tag{3.68}$$

Almansi 应变的本构速度与物体的刚体运动无关。

参考文献

[1]殷有泉.固体力学非线性有限元引论. 北京:北京大学出版社、清华大学出版社，1987
[2]匡震邦.非线性连续介质力学基础. 西安:西安交通大学出版社，1989
[3]W. Prager. Introduction to mechanics of continua. Boston:Ginn and Company, 1961
[4]Y.C. Fung. Foundations of solid mechanics. Prentice-Hall Inc., Englewood Cliffs, N.J., 1965
[5]M.A. Biot. Mechanics of incremental Deformation. New York:John Wiley & Son, Inc. 1965
[6]吕和祥，蒋和洋.非线性有限元. 北京:化学工业出版社， 1992

第4章

应力理论和动力学基本方程

上一章讨论了有限变形情况下变形和应变的度量，本章将在直角坐标系中讨论大变形情况下应力的度量和动力学平衡条件的表述。正交曲线坐标系下的应力理论和动力学基本方程可参考专著或教材[1]~[3]。

4.1 Euler 应力张量

在外力作用下 t 时刻处于平衡的物体，设想沿某截面把处于现时构形中的物体切开。由于整个物体处于平衡状态，切开后的任意部分仍处于平衡状态。为保证切开后的物体仍处于平衡状态，截面上必须存在一定的内力。称这一显示内力的方法为截面法（如图 4.1 所示）。在该假想截面上取一有向微面元 $n_i\Delta A$，ΔA 为其面积，n_i 为该面元的单位法线方向矢量。作用在有向面元上的内力为 ΔT_i，它是 $n_i\Delta A$ 的函数，它能反映一点的受力特性。称

$$t_i^{(n)} = \lim_{\Delta A \to 0} \frac{\Delta T_i}{\Delta A} = \frac{\mathrm{d}T_i}{\mathrm{d}A} \tag{4.1}$$

为应力矢量。由于上述描述是在现时构形中讨论的，采用的是 Euler 方法，因此称 $t_i^{(n)}$ 为 Euler 应力矢量。

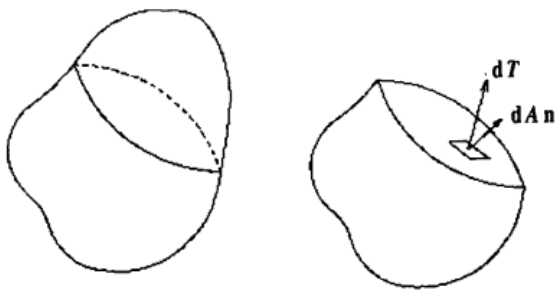


图 4.1 截面法

考虑法线平行于 x_k 坐标轴，且沿其正向的微元面，即 $n_i = \delta_{ki}$。记作用在该微元面上的应力矢量为 $\boldsymbol{t}^k$。定义 $\boldsymbol{t}^k$ 在 x_l 方向的投影为应力张量 $\boldsymbol{\tau}$ 的分量 τ_{kl}，即

$$\tau_{kl} = \boldsymbol{t}^k \cdot \boldsymbol{i}_l \tag{4.2}$$

沿 x_k 坐标轴负向的微元面的应力矢量为 $-\boldsymbol{t}^k$，定义，

$$\boldsymbol{\tau}_{kl}=(-\boldsymbol{t}^{k})\cdot(-\boldsymbol{i}_{l})=\boldsymbol{t}^{k}\cdot\boldsymbol{i}_{l} \tag{4.3}$$

称 $\boldsymbol{\tau}$ 为 Euler 或 Cauchy 应力张量。当 $k=l$ 时，称 $\tau_{\overline{kk}}$ 为正应力，当 $k\neq l$ 时，称 τ_{kl} 为剪应力。这一规定使得拉伸的应力分量始终为正，压缩的压力分量始终为负（见图 4.2）。

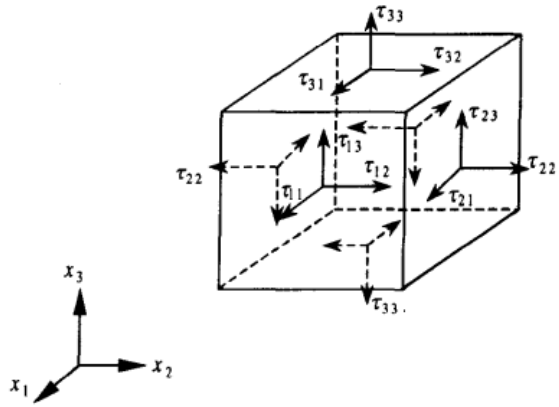


图 4.2 应力分量命名法

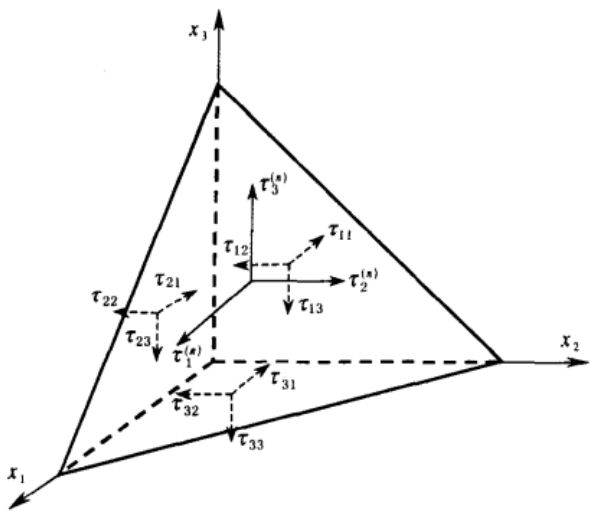


图 4.3 斜截面上的应力矢量

现在我们进一步讨论 Euler 应力张量和 Euler 应力矢量的关系。过现时构形中一点作四面体，其中三个面平行于坐标面（见图 4.3）。设斜截面的面积为 $\mathrm{d}A$，单位法线方向矢量为 $\boldsymbol{n}_i$，作用在该面元上的力为 $\boldsymbol{\tau}^{(n)}\mathrm{d}A$。斜截面有向面元 $\mathrm{d}A\boldsymbol{n}$ 在与 x_k 垂直坐标面上的投影面积为 $\mathrm{d}A\cdot\boldsymbol{n}_k$，作用在该投影面上的 Euler 应力矢量为

$$\boldsymbol{t}_i^{k}=-\tau_{\overline{ki}}\mathrm{d}An_{\overline{k}}$$

设物体单位质量的体积力为 $\boldsymbol{f}$，质点的加速度 $\boldsymbol{a}$，则由四面体的平衡方程可得

$$t_i^{(n)}\mathrm{d}A-\tau_{ki}\mathrm{d}An_k+\rho(f_i-a_i)\mathrm{d}V=0$$

式中，$\mathrm{d}V$ 为四面体的体积，它为比 $\mathrm{d}A$ 更小的小量，在上式中可忽略不计，则

$$t_i^{(n)}=\tau_{ki}n_k \tag{4.4}$$

由上式可以看出，用垂直于坐标轴的三个面上的应力矢量的九个分量 τ_{ij}，可以求得任意一个截面上的 Euler 应力矢量，也就是说，用 τ_{ij} 可以完全刻画该点的应力状态。根据微元体关于力矩的平衡方程，还可以证明 Euler 应力张量是对称的，即

$$\tau_{ij} = \tau_{ji} \tag{4.5}$$

因此，Euler 应力张量有六个独立分量。Euler 应力是定义在现时构形中的单位面积上的力，是与变形相关的真实应力。

4.2　Lagrange 应力张量和 Kirchhoff 应力张量

设物体的初始状态为平衡状态，在外力作用下，或是处于新的平衡状态，或是产生运动。但这种新的平衡或运动都是在物体产生变形以后发生的，变形或运动同样会影响到外力的作用和物体的响应特性。因此研究物体在外力作用下的平衡或运动，原则上应在现时构形中来讨论，也就是说，采用 Euler 应力比较自然。但在固体的有限变形理论中，物体的初始构形是已知的，而现时构形是未知的，必须在问题解出后才能被确定，且固体力学的本构关系在某些情况下也是用 Lagrange 方法给出的，因此，采用 Euler 描述并不方便。本节讨论用 Lagrange 描述法定义的 Lagrange 应力张量和 Kirchhoff 应力张量。

4.2.1　Lagrange 应力张量

Euler 应力是根据现时构形中面元 ΔA 和作用其上的力元矢量 ΔT_i 定义的。假定 ΔA 在初始构形中相应的面元为 ΔA_0，单位法向矢量为 N_i。定义

$$t_i^{*(N)} = \lim_{\Delta A_0 \to 0} \frac{\Delta T_i}{\Delta A_0} = \frac{\mathrm{d}T_i}{\mathrm{d}A_0} \tag{4.6}$$

类似地，取 $N_i = \delta_{ij}$，可定义如下张量，

$$\Sigma_{ij} = t_j^{*(i)} \qquad i = 1,2,3 \quad j = 1,2,3 \tag{4.7}$$

称张量 Σ_{ij} 为拉格朗日（Lagrange）应力张量，也称为名义应力张量或第一类皮奥拉-克希荷夫（Piola-Kirchhoff）应力张量。类似于 Euler 应力，由四面体的平衡条件可得

$$t_i^{*(N)} = \Sigma_{ji} N_j \tag{4.8}$$

现在我们来推导 Euler 应力张量和 Lagrange 应力张量的关系。根据（4.8）式，作用在任意面积为 $\mathrm{d}A_0$、法向单位矢量为 N_i 的面元上的力矢量为

$$\mathrm{d}T_i = \Sigma_{ji} N_j \mathrm{d}A_0 \tag{4.9}$$

根据假定，它与 Euler 应力和对应面元 $\mathrm{d}An_i$ 的关系是，

$$\mathrm{d}T_i = \tau_{ij} n_j \mathrm{d}A$$

利用初始构形和现时构形中物质面元的关系式（3.19），上式可写为

$$J\tau_{ij} \frac{\partial X_m}{\partial x_j} N_m \mathrm{d}A_0 = \Sigma_{mi} N_m \mathrm{d}A_0$$

注意到上式对任意的有向面元 $\mathrm{d}A_0N_i$ 均成立，因此

$$\Sigma_{ij}=J\frac{\partial X_i}{\partial x_k}\tau_{jk} \tag{4.10}$$

对上式求逆，可得

$$\tau_{ij}=J^{-1}\frac{\partial x_j}{\partial X_k}\Sigma_{ki} \tag{4.11}$$

注意到 Lagrange 应力与现时构型中的力和初始构型中的面元有关，它具有两个特点。第一，它是名义应力，不是真实应力；第二，Lagrange 应力一般是不对称的，因此在应用中存在一些不便。

4.2.2 Kirchhoff 应力张量

为了便于应用，构造一个相对于初始构形定义的、而且是对称的应力张量是有必要的。将式（4.10）两边同时乘以 $\frac{\partial X_l}{\partial x_j}$，并适当地更换指标可得

$$S_{lm}=\frac{\partial X_l}{\partial x_i}\Sigma_{mi}=J\frac{\partial X_l}{\partial x_i}\frac{\partial X_m}{\partial x_j}\tau_{ij} \tag{4.12}$$

对上式求逆，可得

$$\begin{aligned}\Sigma_{mi}&=\frac{\partial x_i}{\partial X_l}S_{lm}\\ \tau_{ij}&=J^{-1}\frac{\partial x_i}{\partial X_l}\frac{\partial x_j}{\partial X_m}S_{lm}\end{aligned} \tag{4.13}$$

称 S_{ij} 为克希荷夫（Kirchhoff）应力张量，也叫第二类 Piola-Kirchhoff 应力张量，或称为伪应力张量。

现在我们来分析 Kirchhoff 应力张量的力学含义。类似于 Euler 应力，假想作用在初始构形中面元 $\mathrm{d}A_0N_i$ 上的力元矢量为 $\mathrm{d}T_i^0$，它与 Kirchhoff 应力的关系可表示为

$$\mathrm{d}T_i^0=S_{ij}N_j\mathrm{d}A_0 \tag{4.14}$$

把（4.12）式代入（4.14）式，可得

$$\mathrm{d}T_i^0=J\frac{\partial X_i}{\partial x_l}\frac{\partial X_j}{\partial x_m}\tau_{lm}N_j\mathrm{d}A_0$$

利用初始构形和现时构形中物质面元的关系式（3.19），上式可写为

$$\mathrm{d}T_i^0=\frac{\partial X_i}{\partial x_l}\tau_{lm}n_m\mathrm{d}A=\frac{\partial X_i}{\partial x_l}\mathrm{d}T_l \tag{4.15}$$

即

$$\mathrm{d}T_i=\frac{\partial x_i}{\partial X_j}\mathrm{d}T_j^0 \tag{4.16}$$

式中，$\mathrm{d}T_i$ 为现时构形中物质面元 $n_i\mathrm{d}A$ 上作用的真实的力元，它与由 Kirchhoff 应力定义的名义力元 $\mathrm{d}T_i^0$ 间的变换关系同线元的变化关系完全相同。真实的力元可以看成是由 Kirchhoff 应

力定义的力元经与变形相同的方式被“拉长和转动”后得到的。虽然 Kirchhoff 应力定义的方式不是很自然，但它的物理意义是明确的。在下面的分析中我们可以看到，它是与 Green 应变能量共轭的。此外，Kirchhoff 应力还具有一个十分重要的性质，那就是当物体作刚体运动时，在空间固定的坐标系内 Kirchhoff 应力分量保持不变。

例 4-1　如图 4.4 所示的一杆件，初始构形中长为 L，截面积为 A，在轴向外力 P 作用下，现时构形中长为 l，截面积为 a。试求 Euler 应力，Lagrange 应力和 Kirchhoff 应力。

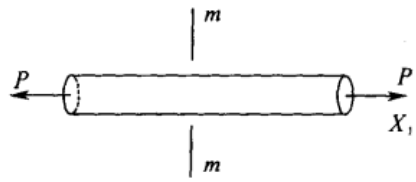


图 4.4　轴向受力杆件

在现时构形中，假想从 m-m 处切开，在 m-m 截面上的应力矢量为 $\boldsymbol{t}^{(1)}$，则

$$\boldsymbol{t}^{(1)}=\frac{\boldsymbol{P}}{a}$$

由此可以求出应力分量，

$$\tau_{11}=P/a \quad \tau_{12}=0 \quad \tau_{13}=0$$

若在初始构形中讨论这一问题，则与 m-m 对应的 M-M 截面上的应力矢量为

$$\boldsymbol{t}^{*(1)}=\frac{\boldsymbol{P}}{A}$$

由此可以求出 Lagrange 应力分量

$$\Sigma_{11}=P/A \quad \Sigma_{12}=0 \quad \Sigma_{13}=0$$

空间坐标 x_1 和物质坐标 X_1 存在如下关系

$$x_1=\frac{l}{L}X_1$$

则

$$S_{1m}=\frac{\partial X_1}{\partial x_k}\Sigma_{mk}=\frac{L}{l}\delta_{k1}\Sigma_{mk}=\frac{L}{l}\Sigma_{m1}$$

显然有

$$S_{11}=\frac{L}{l}\frac{P}{A} \qquad S_{12}=0 \qquad S_{13}=0$$

其它应力分量均为零。

4.3　微分形式的运动方程

本节分析 Euler 应力、Lagrange 应力和 Kirchhoff 应力表示的运动方程。

4.3.1　Euler 应力表示的运动方程

在外荷载作用下，物体发生运动和变形，并在变形后达到运动平衡。因此采用现时构形的 Euler 应力描述物体的运动状态是合适的。设变形体在现时构形中占据的区域为 V，对应的边界为 Γ。Γ 由两部分 Γ_t 和 Γ_u 组成，在 Γ_t 上给定了力的边界条件，在 Γ_u 上给定了位移边界条件，两类边界应当满足，

$$\Gamma_t \cap \Gamma_u = \phi \qquad \Gamma_t \cup \Gamma_u = \Gamma \tag{4.17}$$

在现时构形中，平行于坐标面切出一个微小长方体，三个坐标方向的长度分别为 dx_1、dx_2 和 dx_3，如图 4.5 所示，设微元体内单位体积的体积力为 f_i，加速度为 a_i。现在分析微元体的平衡，显然垂直 x_1 坐标轴的两个面元的力元矢量和为

$$T_i^{(1)} = -\tau_{1i}dx_2dx_3 + \left(\tau_{1i} + \frac{\partial \tau_{1i}}{\partial x_1}dx_1\right)dx_2dx_3$$

即

$$T_i^{(1)} = \frac{\partial \tau_{1i}}{\partial x_1}dx_1dx_2dx_3 = \frac{\partial \tau_{1i}}{\partial x_1}dV$$

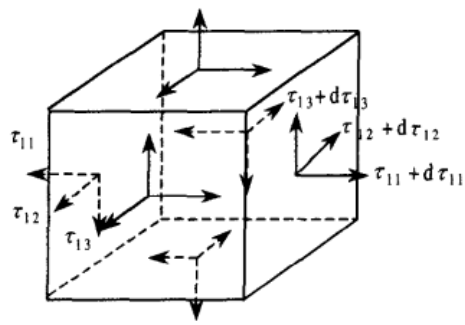


图4.5 微元体

同理有

$$T_i^{(2)} = \frac{\partial \tau_{2i}}{\partial x_2}dx_1dx_2dx_3 = \frac{\partial \tau_{2i}}{\partial x_2}dV$$

$$T_i^{(3)} = \frac{\partial \tau_{3i}}{\partial x_3}dx_1dx_2dx_3 = \frac{\partial \tau_{3i}}{\partial x_3}dV$$

体积力和惯性力的力矢量为

$$T_i^{(b)} = f_i dV - a_i \rho dV$$

根据力的平衡条件应当有

$$T_i^{(1)} + T_i^{(2)} + T_i^{(3)} + T_i^{(b)} = 0$$

即在 V 内有

$$\frac{\partial \tau_{ji}}{\partial x_j} + f_i - \rho a_i = 0 \tag{4.18}$$

根据内应力矢量和外力矢量相等，可得出在 Γ_t 上力的边界条件为

$$\tau_{ij}n_j = \bar{t}_i \tag{4.19}$$

式中，n_j 为力边界的单位外法线方向矢量，$\bar{t}_i$ 为表面力载荷矢量。在 Γ_u 上，位移的边界条件可以表示为

$$u_i = \bar{u}_i \tag{4.20}$$

从形式上看，式（4.18）和（4.19）与小变形情况没有区别，然而其实质是不同的。在这里 τ_{ij} 是定义在现时构形中的 Euler 应力，Γ_t 和 Γ_u 为现时构形中的力和位移边界。

当物体运动的加速度为零时，（4.18）式便是静力学的平衡方程，

$$\frac{\partial \tau_{ij}}{\partial x_j} + f_i = 0 \tag{4.21}$$

4.3.2 Lagrange 应力表示的运动方程

上一节的分析中已经提到，对于固体力学问题来说，由于现时构形是待确定的，因此采用 Euler 描述并不总是很方便，必须在初始构形中给出对现时构形运动的描述（即 Lagrange

描述）。设现时构形中物体占据的区域 V 与边界 Γ_t 和 Γ_u，分别对应初始构形中的区域 V_0 与边界 Γ_t° 和 Γ_u°。记在初始构形中单位体积的体积力为 f_i°，单位面积的面分布力为 $\bar{t}_i^\circ$。假设物质体元 $\mathrm{d}V$ 所受的体力和力边界上的面元所受的面力在变形过程保持不变，即

$$f_i^\circ \mathrm{d}V_0 = f_i \mathrm{d}V \tag{4.22}$$

$$\bar{t}_i^\circ \mathrm{d}A_0 = \bar{t}_i \mathrm{d}A \tag{4.23}$$

一般地，称满足（4.22）和（4.23）式的载荷为保守力系。把 Lagrange 应力和 Euler 应力间的关系式（4.11）式代入运动方程（4.18）式，则有

$$\frac{\partial(J^{-1}\dfrac{\partial x_j}{\partial X_k}\Sigma_{ki})}{\partial x_j} + \frac{1}{J}f_i^\circ - \rho a_i = 0$$

两边同时乘以 J，方程左端第一项为

$$J\frac{\partial(J^{-1}\dfrac{\partial x_j}{\partial X_k}\Sigma_{ki})}{\partial x_j} = J\frac{\partial J^{-1}}{\partial x_j}\frac{\partial x_j}{\partial X_k}\Sigma_{ki} + J\cdot J^{-1}\frac{\partial^2 x_j}{\partial X_k \partial x_j}\Sigma_{ki} + J\cdot J^{-1}\frac{\partial x_j}{\partial X_k}\frac{\partial \Sigma_{ki}}{\partial x_j}$$

注意到

$$\frac{\partial J^{-1}}{\partial x_j}\frac{\partial x_j}{\partial X_k} = \frac{\partial J^{-1}}{\partial X_k} \qquad \frac{\partial^2 x_j}{\partial X_k \partial x_j} = 0 \qquad \frac{\partial x_j}{\partial X_k}\frac{\partial \Sigma_{ki}}{\partial x_j} = \frac{\partial \Sigma_{ki}}{\partial X_k}$$

方程第一项可简化为

$$J\frac{\partial(J^{-1}\dfrac{\partial x_j}{\partial X_k}\Sigma_{ki})}{\partial x_j} = J\frac{\partial J^{-1}}{\partial X_k}\Sigma_{ki} + \frac{\partial \Sigma_{ki}}{\partial X_k}$$

容易证明

$$J\frac{\partial J^{-1}}{\partial X_k} \equiv 0$$

同时注意到 $J\rho = \rho_0$，原方程可表示为

$$\frac{\partial \Sigma_{ki}}{\partial X_k} + f_i^0 - \rho_0 a_i = 0 \tag{4.24}$$

上式即为基于初始构形、以 Lagrange 应力描述的运动方程。要注意的是，式（4.18）中的 a_i 为空间坐标的函数，上式中的 a_i 为物质坐标的函数。

根据 Lagrange 应力的定义和（4.23）式的假设，基于初始构形、用 Lagrange 应力描述的力边界条件为

$$\Sigma_{ki}N_k = \bar{t}_i^0 \tag{4.25}$$

4.3.3　Kirchhoff 应力表示的运动方程

把式（4.13）式代入（4.24）式和（4.25）式，得 Kirchhoff 应力描述的运动方程

$$\frac{\partial\left(\frac{\partial x_i}{\partial X_l}S_{lk}\right)}{\partial X_k}+f_i^0-\rho_0 a_i=0 \tag{4.26}$$

和力边界条件

$$S_{lk}\frac{\partial x_i}{\partial X_l}N_k=\overline{t}_i^0 \tag{4.27}$$

将变形梯度用位移分量表示，则（4.26）式可化为

$$\frac{\partial}{\partial X_k}(S_{ik}+\frac{\partial u_i}{\partial X_l}S_{lk})+f_i^0-\rho_0 a_i=0 \tag{4.28}$$

由上式可以看出大变形情况下运动方程的几何效应是通过位移梯度的形式表现出来，显然，刚体转动对运动方程有很大的影响。

请读者注意，式（4.24）～（4.28）虽然是基于初始构形给出的，但描述的仍然是现时构形的平衡和力的边界条件，而不是初始构形的平衡。

4.4 积分形式的运动方程——虚功方程

上节我们介绍了大变形条件下微分形式的运动方程及其对应的边界条件，本节将介绍积分形式的运动方程，即虚功方程。虚功方程是位移法有限单元分析的理论基础，在有限元分析中具有十分重要的意义。

4.4.1 Euler 应力表示的虚功方程

首先建立空间描述的虚功方程，即现时构形中的虚功方程。在平衡位置考虑一无限小位移变分(虚位移)δu_i，δu_i在V内单值连续，且$u_i+\delta u_i$为几何可能位移，则$u_i+\delta u_i$在给定位移边界Γ_u上满足位移边界条件，即

$$u_i+\delta u_i=\overline{u}_i \tag{4.29}$$

因为u_i为真实位移，它也满足位移边界条件，因此在Γ_u上有

$$\delta u_i=0 \tag{4.30}$$

外力在虚位移上所做的虚功可以表示为

$$\delta w=\int_V f_i\,\delta u_i\mathrm{d}V+\int_{\Gamma_t}\overline{t}_i\,\delta u_i\mathrm{d}\Gamma \tag{4.31}$$

将边界条件(4.19)式代入上式，同时注意到(4.17)和(4.30)式，上式第二个积分中的积分域Γ_t可改为Γ，

$$\int_{\Gamma_t}\overline{t}_i\,\delta u_i\mathrm{d}\Gamma=\int_{\Gamma_t}\tau_{ji}n_j\,\delta u_i\mathrm{d}\Gamma=\int_{\Gamma}\tau_{ji}n_j\,\delta u_i\mathrm{d}\Gamma$$

根据散度定理有

$$\int_{\Gamma_t}\overline{t}_i\,\delta u_i\mathrm{d}\Gamma=\int_V\frac{\partial(\tau_{ji}\,\delta u_i)}{\partial x_j}\mathrm{d}V \tag{4.32}$$

而

$$\frac{\partial(\tau_{ji}\delta u_i)}{\partial x_j}=\frac{\partial \tau_{ji}}{\partial x_j}\delta u_i+\tau_{ji}\frac{\partial(\delta u_i)}{\partial x_j}=\frac{\partial \tau_{ji}}{\partial x_j}\delta u_i+\tau_{ji}\delta\frac{\partial u_i}{\partial x_j}=\frac{\partial \tau_{ji}}{\partial x_j}\delta u_i+\tau_{ji}\delta\varepsilon_{ij} \tag{4.33}$$

式中，

$$\varepsilon_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{4.34}$$

(4.33)式的推导中，利用了 Euler 应力张量的对称性。注意，ε_{ij} 为定义在现时构形中的无限小应变，与上一章提到的 Cauchy 形式相似，但有实质性区别。将(4.33)式代入(4.32)式，再代入(4.31)式，利用微分形式的运动方程(4.18)式，可得

$$\delta w=\int_V(\rho a_i\,\delta u_i+\tau_{ij}\,\delta\varepsilon_{ij})\mathrm{d}V \tag{4.35}$$

于是，我们就得到了空间描述的虚功方程，

$$\int_V\tau_{ij}\delta\varepsilon_{ij}\mathrm{d}V=\int_V(f_i-\rho a_i)\delta u_i\mathrm{d}V+\int_{\Gamma_t}\bar{t}_i\,\delta u_i\mathrm{d}\Gamma \tag{4.36}$$

式(4.36)有明显的物理意义，它表示内力所做的虚功（方程左端）等于外力(包括惯性力)所做的虚功。如果虚功方程的推导中用速度场的变分 δv_i 代替位移场的变分，类似地可以导出虚功率方程，

$$\int_V\tau_{ij}\delta\dot{\varepsilon}_{ij}\mathrm{d}V=\int_V(f_i-\rho a_i)\delta v_i\mathrm{d}V+\int_{\Gamma_t}\bar{t}_i\delta v_i\mathrm{d}\Gamma \tag{4.37}$$

由式(4.36)可以看出，Euler 应力张量 τ_{ij} 和定义在现时构形中的无限小应变　存在能量共轭关系。应当注意的是，式(4.36)和(4.37)的推导过程中没有涉及材料的性质，因此，它们对于任意材料均成立。

4.4.2　Lagrange 应力表示的虚功方程

现在我们来讨论物质描述的虚功方程，假设变形体承受保守力系作用，则

$$\int_V(f_i-\rho a_i)\delta u_i\mathrm{d}V=\int_{V_0}(f_i^0-\rho_0 a_i)\delta u_i\mathrm{d}V_0 \tag{4.38}$$

$$\int_{\Gamma_t}\bar{t}_i\,\delta u_i\mathrm{d}\Gamma=\int_{\Gamma_t^0}\bar{t}_i^0\,\delta u_i\mathrm{d}\Gamma_0 \tag{4.39}$$

将(4.11)代入(4.36)式的左边并注意到(3.14)式，可得

$$\int_V\tau_{ij}\delta\varepsilon_{ij}\mathrm{d}V=\int_V\tau_{ij}\delta\left(\frac{\partial u_i}{\partial x_j}\right)\mathrm{d}V=\int_V J^{-1}\frac{\partial x_j}{\partial X_k}\Sigma_{ki}\delta\left(\frac{\partial u_i}{\partial x_j}\right)\mathrm{d}V=\int_{V_0}\Sigma_{ki}\delta\left(\frac{\partial u_i}{\partial X_k}\right)\mathrm{d}V_0 \tag{4.40}$$

将式(4.38)、(4.39)和(4.40)代入(4.36)式可得 Lagrange 应力张量表示的积分形式运动方程，

$$\int_{V_0}\Sigma_{ji}\delta\left(\frac{\partial u_i}{\partial X_j}\right)\mathrm{d}V_0=\int_{V_0}(f_i^0-\rho_0 a_i)\delta u_i\mathrm{d}V_0+\int_{\Gamma_t^0}\bar{t}_i^0\delta u_i\mathrm{d}\Gamma_0 \tag{4.41}$$

由上式可以看出，Lagrange 应力张量 Σ_{ij} 和相对于初始构形定义的位移梯度 $\dfrac{\partial u_j}{\partial X_i}$ 在能量上共轭。

4.4.3 Kirchhoff 应力表示的虚功方程

将(4.13)代入(4.36)式的左边，同样利用(3.14)式可得

$$\begin{aligned}\int_V \tau_{ij}\delta\varepsilon_{ij}\mathrm{d}V &= \int_V (J^{-1}\frac{\partial x_i}{\partial X_l}\frac{\partial x_j}{\partial X_m}S_{lm})\frac{1}{2}\left[\delta(\frac{\partial u_i}{\partial x_j})+\delta(\frac{\partial u_j}{\partial x_i})\right]\mathrm{d}V \\ &= \int_{V_0}\frac{1}{2}S_{lm}\delta(\frac{\partial x_k}{\partial X_l}\frac{\partial x_k}{\partial X_m})\mathrm{d}V_0 = \int_{V_0} S_{lm}\delta E_{lm}\mathrm{d}V_0\end{aligned} \tag{4.42}$$

将式(4.38)、(4.39)和(4.42)代入(4.36)式，可得 Kirchhoff 应力张量表示的积分形式运动方程，

$$\int_{V_0} S_{ij}\,\delta(E_{ij})\mathrm{d}V_0 = \int_{V_0}(f_i^0-\rho_0 a_i)\,\delta u_i\mathrm{d}V_0 + \int_{\Gamma_t^0}\overline{t}_i^0\,\delta u_i\mathrm{d}\Gamma_0 \tag{4.43}$$

上式说明，Kirchhoff 应力 S_{ij} 和 Green 应变 E_{ij} 在能量上共轭。

4.5 应力率

上一章变形率分析中指出， 质点 P 邻域内任意点相对 P 点的运动可以用速度梯度刻画。而速度梯度又可分解成对称的变形率张量 V_{ij} 和反对称的旋率张量 Ω_{ij}。当 P 点的邻域作瞬时刚体运动，即绕 P 点转动时，变形率张量为零，反映了格林应变对时间的物质导数。由于 Almansi 应变的物质导数不仅和应变率张量有关，而且与表征瞬时刚体转动的旋率张量 Ω_{ij} 有关，不适合直接应用于本构描述，必须将 Almansi 应变的物质导数减去刚体运动引起的部分，定义 Almansi 应变的本构速率。

同样，在弹塑性等与时间或历史有关的本构关系中，我们也经常用到应力的变化率。本构关系中的应力变化率也必须是与刚体运动无关的。下面我们来分析瞬时刚体运动引起欧拉应力和克希荷夫应力的变化。

4.5.1 欧拉应力率和焦曼应力率

考虑物质点 P 的一个邻域，假设该邻域绕 P 点作刚体转动。以 P 点为原点建立坐标系 $o'x_1'x_2'x_3'$，该坐标系跟随邻域中的物质点一起作刚体转动，在 t 时刻坐标系 $o'x_1'x_2'x_3'$ 和固定的直角坐标系 $ox_1x_2x_3$ 一致。P 点邻域中任意点 Q 在坐标系 $o'x_1'x_2'x_3'$ 中的坐标分量 $\mathrm{d}x_i'$ 不因瞬时刚体转动而改变，根据上一章的分析，在 $ox_1x_2x_3$ 坐标系中的坐标分量 $\mathrm{d}x_i$ 以速率

$$\mathrm{d}v_i = -\Omega_{ij}\mathrm{d}x_j = -e_{ijk}\Omega_k\mathrm{d}x_j \tag{4.44}$$

变化。注意 t 时刻有 $\mathrm{d}x_i = \mathrm{d}x_i'$。

$$\mathrm{d}v_i = -e_{ijk}\Omega_k\mathrm{d}x_j' \tag{4.45}$$

$t+\mathrm{d}t$ 时刻的 $\mathrm{d}x_i$ 可以表示为

$$\mathrm{d}x_i = \mathrm{d}x_i' + \mathrm{d}v_i\mathrm{d}t = (\delta_{ij} - e_{ijk}\Omega_k\mathrm{d}t)\mathrm{d}x_j' \tag{4.46}$$

记

$$R_{ij} = \delta_{ij} - e_{ijk}\Omega_k\mathrm{d}t \tag{4.47}$$

则 $t+\mathrm{d}t$ 时刻坐标系 $o'x_1'x_2'x_3'$ 和 $ox_1x_2x_3$ 间的坐标变换关系为

$$\mathrm{d}x_i = R_{ij}\mathrm{d}x_j' \tag{4.48}$$

在运动坐标系 $o'x_1'x_2'x_3'$ 内定义欧拉应力率

$$\overset{\nabla}{\tau}_{ij}(t) = \lim_{\Delta t\to 0}\frac{1}{\Delta t}[\tau_{i'j'}(t+\Delta t) - \tau_{i'j'}(t)] \tag{4.49}$$

当 P 点邻域 t 时刻作瞬时刚体运动时，

$$\overset{\nabla}{\tau}_{ij}(t) = 0 \tag{4.50}$$

在固定坐标系 $ox_1x_2x_3$ 中，$t+\mathrm{d}t$ 时刻质点 P 的欧拉应力可以用物质导数表示为

$$\tau_{ij}(t+\mathrm{d}t) = \tau_{ij}(t) + \dot{\tau}_{ij}(t)\mathrm{d}t \tag{4.51}$$

根据张量的坐标变换关系，$\tau_{i'j'}(t+\mathrm{d}t)$ 和 $\tau_{ij}(t+\mathrm{d}t)$ 存在如下关系

$$\tau_{i'j'}(t+\mathrm{d}t) = R_{pi}R_{qj}\tau_{pq}(t+\mathrm{d}t) \tag{4.52}$$

将式（4.47）和（4.51）代入上式，并化简后可得到

$$\tau_{i'j'}(t+\mathrm{d}t) = \tau_{ij}(t) + (\dot{\tau}_{ij}(t) - \tau_{ip}\Omega_{pj} - \tau_{jp}\Omega_{pi})\mathrm{d}t \tag{4.53}$$

注意到

$$\tau_{ij}(t) = \tau_{i'j'}(t) \tag{4.54}$$

则

$$\lim_{\Delta t\to 0}\frac{\tau_{i'j'}(t+\Delta t) - \tau_{i'j'}(t)}{\Delta t} = \dot{\tau}_{ij}(t) - \tau_{ip}\Omega_{pj} - \tau_{jp}\Omega_{pi} \tag{4.55}$$

比较上式和（4.49）式，有

$$\overset{\nabla}{\tau}_{ij}(t) = \dot{\tau}_{ij}(t) - \tau_{ip}\Omega_{pj} - \tau_{jp}\Omega_{pi} \tag{4.56}$$

由上式可见，当 P 点邻域在 t 时刻作瞬时刚体运动时，$\dot{\tau}_{ij}(t)\neq 0$，即关于刚体转动不具有不变性，不适宜直接应用于增量本构关系。而按式（4.56）定义的应力率是不受刚体转动影响的客观张量，适合于本构描述，我们称之为焦曼（Jaumann）应力率。

4.5.2　克希荷夫应力率

根据式（4.12），克希荷夫应力的物质导数可以表示为

$$\dot{S}_{ij} = \dot{J}\frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_l}\tau_{kl} + J[(\frac{\partial X_i}{\partial x_k})^{\bullet}\frac{\partial X_j}{\partial x_l}\tau_{kl} + \frac{\partial X_i}{\partial x_k}(\frac{\partial X_j}{\partial x_l})^{\bullet}\tau_{kl} + \frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_l}\dot{\tau}_{kl}] \tag{4.57}$$

利用（3.65）式，

$$J(\frac{\partial X_i}{\partial x_k})^{\bullet}\frac{\partial X_j}{\partial x_l}\tau_{kl} = -J\frac{\partial v_p}{\partial x_k}\frac{\partial X_i}{\partial x_p}\frac{\partial X_j}{\partial x_l}\tau_{kl} \tag{4.58}$$

$$J\frac{\partial X_i}{\partial x_k}(\frac{\partial X_j}{\partial x_l})^{\bullet}\tau_{kl} = -J\frac{\partial v_p}{\partial x_l}\frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_p}\tau_{kl} \tag{4.59}$$

利用（3.15）式

$$\dot{J}=(\frac{\rho_0}{\rho})^{\cdot}=-\frac{\rho_0}{\rho^2}\dot{\rho} \tag{4.60}$$

将（3.49）式代入上式，可得

$$\dot{J}=\frac{\rho_0}{\rho}\frac{\partial v_p}{\partial x_p}=J\frac{\partial v_p}{\partial x_p} \tag{4.61}$$

将式（4.58）、（4.59）和（4.61）代入式（4.57），利用 τ_{ij} 的对称性最后得

$$\dot{S}_{ij}=J\frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_l}(\dot{\tau}_{kl}+\tau_{kl}\frac{\partial v_p}{\partial x_p}-\tau_{kp}\frac{\partial v_l}{\partial x_p}-\tau_{lp}\frac{\partial v_k}{\partial x_p}) \tag{4.62}$$

将（3.55）式代入上式，可得

$$\dot{S}_{ij}=J\frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_l}\left[(\tau_{kl}\frac{\partial v_p}{\partial x_p}-\tau_{kp}V_{lp}-\tau_{lp}V_{kp})+(\dot{\tau}_{kl}+\tau_{kp}\Omega_{lp}+\tau_{lp}\Omega_{kp})\right] \tag{4.63}$$

注意到 Ω_{ij} 为反对称张量，利用式（4.56）有

$$\dot{S}_{ij}=J\frac{\partial X_i}{\partial x_k}\frac{\partial X_j}{\partial x_l}(\overset{\nabla}{\tau}_{kl}+\tau_{kl}\frac{\partial v_p}{\partial x_p}-\tau_{kp}V_{lp}-\tau_{lp}V_{kp}) \tag{4.64}$$

当 P 点邻域的物质元作瞬时刚体运动时，$V_{kl}=0$，$\partial v_p/\partial x_p=0$，$\overset{\nabla}{\tau}_{kl}=0$，于是

$$\dot{S}_{ij}=0 \tag{4.65}$$

即克希荷夫应力的物质导数为零，因此是一个不随刚体转动的客观张量。从以上推导我们还可以看出物理量

$$\overset{\square}{\tau}_{ij}=\dot{\tau}_{ij}+\tau_{ij}\frac{\partial v_p}{\partial x_p}-\tau_{ip}\frac{\partial v_j}{\partial x_p}-\tau_{jp}\frac{\partial v_i}{\partial x_p} \tag{4.66}$$

也是一个不受刚体转动影响的客观量，可以作为应力率的另一种表达式，这就是特斯德尔（Truesdell）定义的应力率。

参考文献

[1] 匡震邦.非线性连续介质力学基础. 西安:西安交通大学出版社， 1989

[2] 殷有泉.固体力学非线性有限元引论. 北京:北京大学出版社、清华大学出版社，1987

[3] J.T. Oden.Finite elements of nonlinear continua. New York:Mc Graw-Hill, Inc. 1972

第5章

本构关系理论

前面几章介绍了适用于所有连续介质的共同原理，导出了运动微分方程和几何方程，但没有涉及具体材料的物理特性。光靠上述方程不足以解决固体力学问题，还必须建立一组把应力、应变和温度变化等联系起来的方程，它反映了材料的固有物理特性，称为本构关系。由于具体材料物质结构的复杂性和变形机理的多样性，要通过理论分析得出一个对任何连续介质和各种工作条件下都适用的本构关系是不可能的。一般的做法是，先根据热力学基本定律确定本构方程的基本框架，再配合适当的材料试验测定必要的材料特性参数，从而得到某类材料在特定工作条件下便于实际应用的本构关系。

5.1 本构方程的基本原理

自然界的物质是丰富多彩的，因此不同的物质需要用不同的本构关系来描述。然而，不同的物质又都属于同一自然界，因此所有物质的本构方程又必须服从一些共同的准则或原理。这些原理对物质的本构方程施加了确定的限制。以下简单介绍本构关系的一些基本原理。详细内容可参见著作[1]~[3]。

一、物质客观性原理

本构方程是由物质性质决定的，它不随观测者的改变而变化，因而作相对运动的两个观测者在做材料试验时应当得到相同的本构关系。换言之，在作相对运动的坐标系中，本构方程应具有相同的形式。

同时，本构方程不因坐标系的选择不同而不同，在各坐标系中均具有相同的形式，本构方程应具有张量的性质。物质客观性原理也称为坐标系无差异原理。

二、确定性（遗传性）原理

确定性原理认为物体中某点 $\boldsymbol{X}$ 的状态恒可由物体内各点的以往运动史惟一确定，而和未来的运动无关。也就是说，表征物体中某点 $\boldsymbol{X}$ 状态的物理量 Θ 应由下式确定：

$$\Theta(\boldsymbol{X},t)=\Re(\boldsymbol{x}(\boldsymbol{X},\tau);\boldsymbol{X},t)$$

上式中，$\tau\in(-\infty,t]$，$\Re$ 是 $\boldsymbol{X}$ 和 t 的普通函数，是函数 $\boldsymbol{x}(\boldsymbol{X},\tau)$ 的泛函数。称 $\Re$ 为本构泛函或响应函数。

三、局部作用原理

物体内诸点的运动对某点 $\boldsymbol{X}$ 的应力或其他物理量的影响，随着离该点距离的增大而减小，特别是对大多数物质，只有 $\boldsymbol{X}$ 点邻近质点的运动才对 $\boldsymbol{X}$ 点的应力有影响，而离开 $\boldsymbol{X}$ 点有限距离的物质点的运动对该点的应力没有影响。换句话说，同一物体的两个不同的运动历史，只

要保证两个运动历史在 $\boldsymbol{X}$ 点邻域内是相同的，则 $\boldsymbol{X}$ 点的应力或其他物理量也是相同的。

物质客观性原理、确定性原理和局部作用原理也称为 Noll 三原则，由此可以导出简单物质的本构关系。

四、物质对称性原理

自然界中的物质一般都存在这样和那样的对称性，本构方程必须能够反映这种对称性。线性弹性力学的理论已证明，理论上应力-应变间存在 81 个弹性常数。由于应力和应变张量均为对称张量，独立的弹性常数下降到 21 个，正交各向异性物质具有三个相互正交的对称面，可以导出独立的弹性常数为 9 个。横观各向同性材料有 5 个独立的弹性常数，而各向同性材料则只有 2 个独立的弹性常数[3]。

五、记忆减退原理

离现时刻越远的过去的历史对现时刻应力和其他状态物理量的影响越小。蠕变材料等的本构方程都必须满足这一原理。

六、相容性原理

物质的本构关系必须满足一些普遍原理，如质量守恒原理、动量守恒原理、动量矩方程、局部能量守恒原理、熵产率原理等。

七、量纲一致性原理

本构方程中各项的量纲必须一致，同时要特别注意一些特殊函数（如对数函数、指数函数等）的自变量量纲。从事本构关系研究时，对量纲一致性原理必须有足够的重视，否则，即使提出的本构关系能很好地拟合试验数据，但在实际应用中可能会得到完全错误的结果。

5.2 弹性介质的本构关系

物体在外部荷载作用下产生变形和应力，在外力不大的情况下，当外部荷载卸除后，变形和应力也随之消失。变形的这种可恢复性（可逆性）叫作弹性，任何真实介质都或多或少地具有弹性。弹性材料对过去的历史没有记忆，对变形和温度的反应仅限于当前的状态。假设存在一个无应力的自然状态，并限于考虑等温过程，欧拉应力张量 τ_{ij} 和阿耳曼西应变之间存在一一对应关系，

$$\tau_{ij} = D_{ijkl} e_{kl} \tag{5.1}$$

如果四阶材料张量 D_{ijkl} 是应变张量 e_{ij} 和（或）应力张量 τ_{ij} 的函数，则为非线性弹性。如果 D_{ijkl} 是常数张量，则为线性弹性。要注意的是，（5.1）式不是小变形情况的胡克定律，因为（5.1）式中的物理量是相对现时构形定义的。在小变形条件下，τ_{ij} 和 e_{ij} 退化为通常使用的工程应力 σ_{ij} 和无限小应变 ε_{ij}，此时，式（5.1）退化为胡克定律。对于各向同性材料，无论坐标轴怎样选取，D_{ijkl} 的形式保持不变，为各向同性张量，可以用两个独立的常数表示，

$$D_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{5.2}$$

这里 λ 和 μ 称为拉梅（Lame）常数。

为便于应用，通常需要从定义在现时构形中的本构关系导出定义在初始构形中的本构关系，即要导出 Kirchhoff 应力和 Green 应变间的本构关系。根据 Green 应变的定义，

$$E_{ij}=\frac{1}{2}\left(\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j}-\delta_{ij}\right)$$

上式两边同时乘以$\frac{\partial X_i}{\partial x_m}\frac{\partial X_j}{\partial x_n}$，得

$$\frac{\partial X_i}{\partial x_m}\frac{\partial X_j}{\partial x_n}E_{ij}=\frac{1}{2}\left(\frac{\partial X_i}{\partial x_m}\frac{\partial X_j}{\partial x_n}\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j}-\delta_{ij}\frac{\partial X_i}{\partial x_m}\frac{\partial X_j}{\partial x_n}\right)$$

根据复合函数求导法则，上式右端即为 Almansi 应变e_{mn}，于是有

$$e_{mn}=\frac{\partial X_i}{\partial x_m}\frac{\partial X_j}{\partial x_n}E_{ij} \tag{5.3}$$

把上式和 Euler 应力与 Kirchhoff 应力间的关系式（4.13）代入（5.1）式，可得

$$J^{-1}\frac{\partial x_i}{\partial X_l}\frac{\partial x_j}{\partial X_m}S_{lm}=D_{ijkl}\frac{\partial X_m}{\partial x_k}\frac{\partial X_n}{\partial x_l}E_{mn}$$

由上式不难得出用S_{ij}和E_{ij}定义现时构形材料性质的本构关系，

$$S_{ij}=D^0_{ijkl}E_{kl} \tag{5.4}$$

式中，D^0_{ijkl}和D_{ijkl}之间的转换关系为

$$D^0_{mnpq}=J\frac{\partial X_m}{\partial x_i}\frac{\partial X_n}{\partial x_j}\frac{\partial X_p}{\partial x_k}\frac{\partial X_q}{\partial x_l}D_{ijkl} \tag{5.5}$$

$$D_{mnpq}=J^{-1}\frac{\partial x_m}{\partial X_i}\frac{\partial x_n}{\partial X_j}\frac{\partial x_p}{\partial X_k}\frac{\partial x_q}{\partial X_l}D^0_{ijkl} \tag{5.6}$$

对于非线性弹性模型，应力应变间存在非线性关系，D_{ijkl}不再是常数，而是应变e_{ij}或应力τ_{ij}的函数。为便于与线弹性应力应变关系相区别，习惯上将（5.1）式可改写为

$$\tau_{ij}=D^{(S)}_{ijkl}e_{kl} \tag{5.7}$$

$D^{(S)}_{ijkl}$在形式上和D_{ijkl}相同，为材料割线弹性常数。在有限元分析中,(5.7)式一般要写成增量形式，

$$\mathrm{d}\tau_{ij}=D^{(T)}_{ijkl}\mathrm{d}e_{kl} \tag{5.8}$$

$D^{(T)}_{ijkl}$称为切线弹性常数。在岩土工程中，著名的 Duncan-Zhang 模型即为非线性弹性模型，其中$D^{(S)}_{ijkl}$和$D^{(T)}_{ijkl}$为应力的函数。

以上我们直接按照定义建立了应力和应变之间的函数关系，这种方法叫 Cauchy 法。此外我们也可以从建立应变能函数入手来导出本构关系，这种方法我们称之为 Green 法。存在应变能的材料为超弹性材料或 Green 弹性材料。假设物体变形后具有势能（弹性应变能）W，W为应变的标量函数。注意到 Kirchhoff 应力S_{ij}和 Green 应变E_{ij}在能量上是共轭的，因此在应变E_{ij}和应力S_{ij}的一个状态下，给应变一个微小的变化$\mathrm{d}E_{ij}$，相应的应变能的微小变化是，

$$\mathrm{d}W=S_{ij}\mathrm{d}E_{ij}$$

同时注意应变能的微小变化可以表示为

$$dW = \frac{\partial W}{\partial E_{ij}} dE_{ij}$$

比较以上两式，并注意到 δE_{ij} 的任意性，有

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} \tag{5.9}$$

由上式可以导出增量形式的本构方程为

$$dS_{ij} = \frac{\partial^2 W}{\partial E_{ij} \partial E_{kl}} dE_{kl} \tag{5.10}$$

比较（5.8）式和（5.10）式，我们可以类似地得到 Kirchhoff 应力 S_{ij} 和 Green 应变 E_{ij} 间的切线弹性常数，

$$D_{ijkl}^{0(T)} = \partial^2 W / \partial E_{ij} \partial E_{kl} \tag{5.11}$$

如果定义矢量，

$$\boldsymbol{S} = \{S_{11}, S_{22}, S_{33}, S_{23}, S_{31}, S_{12}\}^T$$

$$\boldsymbol{E} = \{E_{11}, E_{22}, E_{33}, 2E_{23}, 2E_{31}, 2E_{12}\}^T$$

则应力应变关系可表示为

$$\boldsymbol{S} = \boldsymbol{D}_S^0 \boldsymbol{E} \tag{5.12}$$

$$d\boldsymbol{S} = \boldsymbol{D}_T^0 d\boldsymbol{E} \tag{5.13}$$

对于各向同性线性弹性材料，

$$\boldsymbol{D}_T^0 = \boldsymbol{D}_S^0 = \frac{E^0}{1+\nu^0} \begin{bmatrix} \frac{1-\nu^0}{1-2\nu^0} & \frac{\nu^0}{1-2\nu^0} & \frac{\nu^0}{1-2\nu^0} & & & \\ & \frac{1-\nu^0}{1-2\nu^0} & \frac{\nu^0}{1-2\nu^0} & & & \\ & & \frac{1-\nu^0}{1-2\nu^0} & & & \\ & \text{对称} & & \frac{1}{2} & & \\ & & & & \frac{1}{2} & \\ & & & & & \frac{1}{2} \end{bmatrix} \tag{5.14}$$

式中，E^0 和 ν^0 为弹性常数，在小变形条件下，$E^0 = E$ 为杨氏（Young's）模量，$\nu^0 = \nu$ 称为泊松（Poisson）系数或泊松比。

5.3 应力空间表述的小变形弹塑性本构关系

从材料力学的单轴拉伸试验曲线（见图 5.1）可观察到弹塑性材料的某些特性。从自然状态出发，存在一个屈服极限 σ_y（对应于图 5.1 上的 A 点），当应力低于这个极限应力时，应力

与应变服从弹性体的变形规律，卸载后，试件能恢复到初始状态。但当应力超过这个极限（例如达到 B 点）时，应力与应变之间不再服从弹性规律，在卸除外部作用后，变形仅部分地恢复，另一部分作为塑性变形被保留下来。因此，应力与应变之间不再像弹性那样是单值对应的，应力和应变与变形的历史有关。称 σ_y 为初始屈服应力。

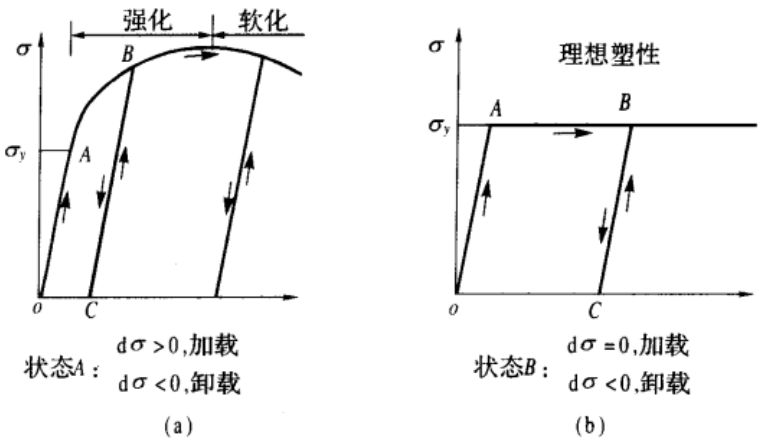


图 5.1　简单应力状态下弹塑性材料

如图 5.1(a)所示，随着塑性变形的出现和发展，材料对外部作用的反应也不同了。例如，从 B 点卸载到达 C 点，然后重新加载，卸载和加载过程都保持弹性，直到 B 点才会出现新的塑性变形，也就是说，屈服极限值可因塑性变形的发展而提高。这种屈服极限值提高的现象叫做强化。同样，屈服极限值也可能随塑性变形的发展而降低，这种现象叫做软化。随塑性变形的发展，屈服极限值变化不大、基本保持为常数的性质叫理想塑性(见图 5.1(b))。要注意的是，有部分工程材料不存在明显的屈服点，工程上通常定义当应变达到 0.2%时的应力为名义屈服应力，并记为 $\sigma_{0.2}$。除可能会产生不可逆变形外，弹塑性材料的另一个重要特点是，应力应变关系和加载速率无关，即所谓速率无关性材料。

由于弹塑性材料对外部作用的反应与变形的历史有关（这可称为历史相关性或路径相关性），本构方程应以增量形式写出。特别地，在塑性状态下（例如图 5.1 中的 B 点），加载和卸载时材料服从不同的规律，这使本构方程的表述要比非线性弹性情况复杂一些。作为简单应力状态的一种推广，通常把应变 ε_{ij} 分成弹性应变 ε_{ij}^{e} 和塑性应变 ε_{ij}^{p} 两部分，即

$$\varepsilon_{ij} = \varepsilon_{ij}^{e} + \varepsilon_{ij}^{p} \tag{5.15}$$

弹性应变 ε_{ij}^{e} 只与当前的应力状态有关，卸载后完全恢复；塑性应变 ε_{ij}^{p} 与加载历史有关，卸载后不能恢复。

5.3.1　应力空间中的屈服条件和屈服面

在复杂应力状态下，材料屈服的描述要比单轴情况复杂得多，建立弹塑性的本构理论，首先需要将上述单向应力状态下建立的概念加以推广。屈服条件的概念就是屈服应力概念的推广。对于应变率较低的等温情形，作为一个经验事实，当应力满足如下条件时材料发生屈服，处于塑性状态，

$$f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)=0 \tag{5.16}$$

其中，$\boldsymbol{\sigma}$ 是六维应力矢量；$\boldsymbol{\sigma}^p$ 是塑性应力矢量，由下面（5.17）式定义。用 $\boldsymbol{\varepsilon}^p$ 表示塑性应变矢量；κ 为内变量，可以是塑性功 w^p、塑性体应变 θ^p 或等效塑性应变 $\bar{\varepsilon}_p$。

$$\boldsymbol{\sigma}^p=\boldsymbol{D}\boldsymbol{\varepsilon}^p \tag{5.17}$$

$$w^p=\int\boldsymbol{\sigma}^T\mathrm{d}\boldsymbol{\varepsilon}^p \tag{5.18}$$

$$\theta^p=\int\boldsymbol{m}^T\mathrm{d}\boldsymbol{\varepsilon}^p=\int\mathrm{d}\varepsilon_{ii}^p,\quad \boldsymbol{m}^T=\{1,1,1,0,0,0\}^T \tag{5.19}$$

$$\bar{\varepsilon}^p=\int((\mathrm{d}\boldsymbol{\varepsilon}^p)^T\mathrm{d}\boldsymbol{\varepsilon}^p)^{1/2} \tag{5.20}$$

通常，称 $f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)$ 为屈服函数，称 $f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)=0$ 为屈服条件或屈服准则。矢量 $\boldsymbol{\sigma}^p$ 和标量 κ 都是标志材料内部结构永久性变化的量，通称为内变量，是加载历史的函数。要特别注意的是，塑性应力矢量 $\boldsymbol{\sigma}^p$ 虽然具有应力的量纲，但并非真实存在，它只是塑性应变矢量 $\boldsymbol{\varepsilon}^p$ 在应力空间中的一种表征，不是产生塑性应变的那部分应力。

从自然状态开始第一次屈服的屈服条件叫初始屈服条件，它可以表示为

$$f=f^*(\boldsymbol{\sigma})=0 \tag{5.21}$$

由于产生了塑性变形，随内变量的增长屈服条件发生了变化，这时的屈服条件叫后继屈服条件。屈服条件（5.21）和（5.16）可以看作是六维应力空间中的超曲面，因而它们也被称为初始屈服面和后继屈服面，通称为屈服面。

对于一个特定的物质质点，它的状态可用应力 $\boldsymbol{\sigma}$ 和内变量 $\boldsymbol{\sigma}^p$、κ 来描述。一般地，使 $f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)<0$ 的状态称为弹性状态，这时对无限小的外部作用的反应是弹性的。使 $f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)=0$ 的状态称为塑性状态，对外部作用的反应是弹塑性的。而 $f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)>0$ 的状态是不存在的。

屈服面随内变量的发展而变化的规律叫做强化规律。人们依据材料的实验资料建立了各种强化模型。目前广泛采用的一种简单的强化模型是等向强化，它假设屈服面作均匀扩大（见图 5.2（a））。这时的后继屈服面仅决定于一个单个参数 κ，如果 $f^*=0$ 是初始屈服面，那么等向强化的后继屈服面可表示为

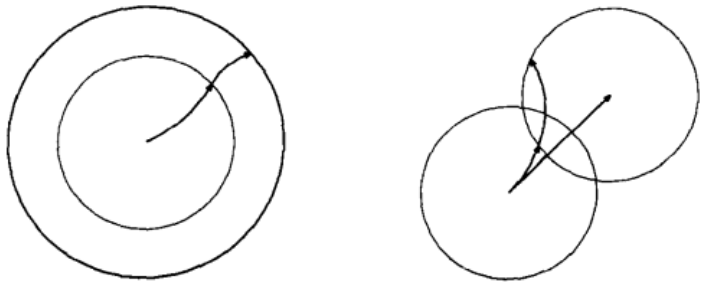
$$f=f^*(\boldsymbol{\sigma})-\mathrm{K}(\kappa)=0 \tag{5.22}$$

其中，函数 K 是标量内变量 κ 的函数，代表屈服面的膨胀。另一种经常使用的强化模型是随动(运动)强化，它假设在塑性变形发展时，屈服面的大小和形状不变，仅是整体地在应力空间中作平动(见图 5.2(b))。以 $\alpha\boldsymbol{\sigma}^p$ 表示屈服面中心的移动矢量，后继屈服面可表示为

$$f=f^*(\boldsymbol{\sigma}-\alpha\boldsymbol{\sigma}^p)=0 \tag{5.23}$$

式中，$\alpha\boldsymbol{\sigma}^p$ 代表应力空间屈服面中心的移动量，称为“背应力”。对于大多数的实际材料，屈服面的强化规律兼有等向强化和随动强化的特点。对既有等向强化，又有运动强化的材料，此时可采用下列屈服面，

$$f^*(\boldsymbol{\sigma}-\alpha\boldsymbol{\sigma}^p)-\mathrm{K}(\kappa)=0 \tag{5.24}$$



(a) 等向强化　　(b) 随动强化

图 5.2　常用强化规律

Bridgman（1928 年）的试验表明，除压力非常高的情况外，对拉压屈服限相同的一般金属，平均应力对屈服的影响甚微。然而，对拉压屈服限不同的材料，或平均应力会引起介质体积明显变化的情况，则不能忽略平均应力对屈服的影响，如某些高强钢、砂子、粘土、粉软岩石等则必须考虑平均应力的影响。

包兴格(Bauschinger)1881 年发现，拉伸（压缩）变形后的金属，当反向压缩（拉伸）时，其屈服应力明显低于拉伸（压缩）时的屈服应力。这一现象通常被称为 Baushinger 效应。

在外部作用过程中，如果在应力空间中应力方向变化不大，使用等向强化规律与实际情况比较符合。由于等向强化模型便于数学处理，应用较为广泛。随动强化规律可以考虑材料的包兴格效应，在循环加载或可能出现反向屈服的问题中，需要使用这种强化模型。

屈服函数有很多，下面仅介绍两个比较著名的屈服函数。

Mises 提出一个二次形式的屈服函数

$$f=\alpha_{klmn}\sigma_{kl}\sigma_{mn}-\aleph^2=0 \tag{5.25}$$

其中，

$$\alpha_{klmn}=\alpha_{lkmn}=\alpha_{klnm}=\alpha_{mnkl}$$

式中共有 21 个独立参数 α_{klmn}，这些参数依赖于参照系的选择，$\aleph$ 称为硬化参数。对于初始屈服面，α_{klmn} 和 $\aleph$ 均为常数，对于后继屈服面，这些参数和塑性应变史有关。对于各向同性弹性体，任何坐标轴都是物质的对称轴，因此可以把三个坐标轴分别选为应力的三个主方向。如果同时假设屈服函数和平均应力没有关系，则由（5.25）式可以导出如下常用的屈服函数，

$$f=\frac{1}{2}\left[(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2\right]-\sigma_y^2=0 \tag{5.26}$$

上式即为各向同性弹塑性体中广泛采用的 Mises 屈服准则。

Tresca 提出了最大剪应力准则，其屈服函数为

$$f=\left[(\sigma_2-\sigma_3)^2-4\tau_y^2\right]\left[(\sigma_3-\sigma_1)^2-4\tau_y^2\right]\left[(\sigma_1-\sigma_2)^2-4\tau_y^2\right]=0 \tag{5.27}$$

一般地说，（5.26）和（5.27）式适合于初始屈服，初始屈服后由于塑性应变的存在，材料已由各向同性转变为各向异性。因此，严格地说，后继屈服面应当采用（5.25）式。

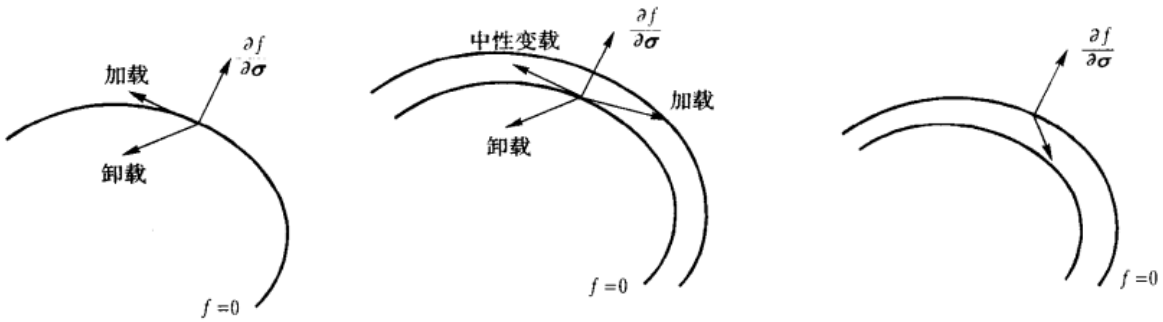
5.3.2　应力空间中的加载、卸载准则

物体初始屈服以后，继续加载的本构方程和卸载时是不同的，因而在复杂应力状态下，确定物体是处于加载过程还是卸载过程是至关重要的。

首先考虑理想塑性材料。由于理想塑性材料屈服面的大小和形状不随内变量发展而变化，后继屈服面和初始屈服面相同，如式（5.21）所示。因此，物质点处于塑性状态的必要条件是该点应力满足（5.21）式。在这样的塑性状态下，材料对所施加应力增量 $\mathrm{d}\boldsymbol{\sigma}$ 有两种不同的反应：一种是有新的塑性变形增量 $\mathrm{d}\boldsymbol{\varepsilon}^p$ 出现，这种情况称为塑性加载（简称加载）；另一种情况是无新的塑性变形产生，即 $\mathrm{d}\varepsilon_{ij}^p=0$，反应是纯弹性的，这种情况叫塑性卸载（简称卸载）。在卸载期间，材料是从塑性状态退回到弹性状态，即应力点离开屈服面向屈服面内侧运动。而在加载期间，材料从一个塑性状态过渡到另外一个塑性状态，应力点始终位于塑性屈服面上，即应力点沿屈服面的切线方向运动。因而理想弹塑性材料的加-卸载准则可表示为

$$f=0,\quad l=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \mathrm{d}\boldsymbol{\sigma}\begin{cases}<0 & \text{卸载}\\ =0 & \text{加载}\end{cases} \tag{5.28}$$

式中，$\dfrac{\partial f}{\partial \boldsymbol{\sigma}}$ 代表应力空间中屈服面 $f=0$ 的外法线方向，如图 5.3(a)所示。



（a）理想弹塑性材料　　（b）强化塑性材料　　（c）软化塑性材料

图 5.3　加-卸载准则

对于强化塑性材料，如图 5.3(b)所示，在加-卸载之间存在一个中间情况，即中性变载，在中性变载期间没有新的塑性变形发生，但应力点保持在屈服面上。因此强化塑性材料的加-卸载准则是，

$$f=0,\quad l=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \mathrm{d}\boldsymbol{\sigma}\begin{cases}<0 & \text{卸载}\\ =0 & \text{中性变载}\\ >0 & \text{加载}\end{cases} \tag{5.29}$$

对于软化塑性材料，如图 5.3（c）所示，加载时屈服面收缩，此时应力增量同卸载时一样，也指向当时屈服面的内侧。因而不能给出一个区别加载和卸载的表达式，也就是说，在应力空间中不能描述软化塑性材料。

5.3.3　应力空间中的关联流动法则和一致性条件

理论和实验证明，对于弹性系数不随内变量的发展而变化的材料，塑性应变增量 $\mathrm{d}\varepsilon_{ij}^p$ 的方向与屈服面的外法线方向一致，即有

$$\mathrm{d}\boldsymbol{\varepsilon}^p=\mathrm{d}\lambda\frac{\partial f}{\partial \boldsymbol{\sigma}} \tag{5.30}$$

其中，$\mathrm{d}\lambda$ 是一个非负的尺度参数，加载时 $\mathrm{d}\lambda>0$，其他情况 $\mathrm{d}\lambda=0$。由于式（5.30）式在几何上表示塑性应变增量矢量 $\mathrm{d}\boldsymbol{\varepsilon}^p$ 与屈服面正交，因此，式（5.30）表述的是应力空间中的关联流动法则，也称为在应力空间中表述的正交法则。

对于强化塑性材料，加载时，屈服面将沿外法线方向向外扩张，应力必须位于扩张后的屈服面上。即加载过程中，$f=0$ 始终满足，则

$$\mathrm{d}f=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma}+\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T\mathrm{d}\boldsymbol{\sigma}^p+\frac{\partial f}{\partial \kappa}\mathrm{d}\kappa=0 \tag{5.31}$$

上式通常称为一致性条件。由于内变量 κ 一般也为塑性应变的函数，上式实际上反映了屈服面发展过程中，微小塑性应变增量和微小应力增量间应满足的关系。

5.3.4　应力空间表述的应力应变关系

在一个无限小的应力增量 $\mathrm{d}\boldsymbol{\sigma}$ 作用下，产生的应变增量 $\mathrm{d}\boldsymbol{\varepsilon}$ 可假设为由弹性部分和塑性部分组成，即

$$\mathrm{d}\boldsymbol{\varepsilon}=\mathrm{d}\boldsymbol{\varepsilon}^e+\mathrm{d}\boldsymbol{\varepsilon}^p \tag{5.32}$$

其中，弹性应变增量 $\mathrm{d}\boldsymbol{\varepsilon}^e$ 与应力增量 $\mathrm{d}\boldsymbol{\sigma}$ 之间满足如下线性关系，

$$\mathrm{d}\boldsymbol{\varepsilon}^e=\boldsymbol{C}\mathrm{d}\boldsymbol{\sigma} \tag{5.33}$$

式中，$\boldsymbol{C}=\boldsymbol{D}^{-1}$ 为柔性系数矩阵。将（5.33）和（5.30）式代入（5.32）式，得

$$\mathrm{d}\boldsymbol{\varepsilon}=\boldsymbol{C}\mathrm{d}\boldsymbol{\sigma}+\frac{\partial f}{\partial \boldsymbol{\sigma}}\mathrm{d}\lambda \tag{5.34}$$

在卸载和中性变载时 $\mathrm{d}\lambda=0$，上式回到增量形式的胡克定律，反应是纯弹性的。在加载时，如果材料是强化的，$\mathrm{d}\lambda>0$，其大小可用一致性条件(5.31)式确定，具体推导如下：

$$\begin{aligned}\mathrm{d}f&=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma}+\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T\mathrm{d}\boldsymbol{\sigma}^p+\frac{\partial f}{\partial \kappa}\mathrm{d}\kappa\\&=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma}+\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T\boldsymbol{D}\mathrm{d}\boldsymbol{\varepsilon}^p+\frac{\partial f}{\partial \kappa}\mathrm{d}\kappa\\&=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma}+\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T\boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}}\cdot\mathrm{d}\lambda+\frac{\partial f}{\partial \kappa}\mathrm{d}\kappa\end{aligned}$$

注意到 κ 一般为塑性应变的函数，则

$$\mathrm{d}\kappa=\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\mathrm{d}\boldsymbol{\varepsilon}^p=\mathrm{d}\lambda\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\frac{\partial f}{\partial \boldsymbol{\sigma}}$$

将上式代入(5.31)可得

$$\mathrm{d}f=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma}+\left[\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T\boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}}+\frac{\partial f}{\partial \kappa}\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\frac{\partial f}{\partial \boldsymbol{\sigma}}\right]\mathrm{d}\lambda=0$$

上式可简写为

$$\mathrm{d}\lambda=\frac{1}{\boldsymbol{A}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathrm{d}\boldsymbol{\sigma} \tag{5.35}$$

其中，

$$A=-\left[\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T \boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}}+\frac{\partial f}{\partial \kappa}\cdot\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\frac{\partial f}{\partial \boldsymbol{\sigma}}\right] \tag{5.36}$$

当 κ 为塑性功 $w^p=\int \boldsymbol{\sigma}^T \mathrm{d}\boldsymbol{\varepsilon}^p$ 时，

$$\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}=\frac{\partial w^p}{\partial \boldsymbol{\varepsilon}^p}=\boldsymbol{\sigma} \tag{5.37}$$

当 κ 为塑性体积应变 $\theta^p=\int \mathrm{d}\varepsilon_{kk}^p$ 时

$$\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}=\frac{\partial \theta^p}{\partial \boldsymbol{\varepsilon}^p}=\frac{\partial \int \boldsymbol{m}^T \mathrm{d}\boldsymbol{\varepsilon}^p}{\partial \boldsymbol{\varepsilon}^p}=\boldsymbol{m} \tag{5.38}$$

当 κ 为等效塑性应变 $\bar{\varepsilon}^p=\int\left((\mathrm{d}\boldsymbol{\varepsilon}^p)^T \mathrm{d}\boldsymbol{\varepsilon}^p\right)^{1/2}$ 时，

由于

$$\mathrm{d}\bar{\varepsilon}^p=\sqrt{(\mathrm{d}\boldsymbol{\varepsilon}^p)^T \mathrm{d}\boldsymbol{\varepsilon}^p}=\frac{(\mathrm{d}\boldsymbol{\varepsilon}^p)^T}{\sqrt{(\mathrm{d}\boldsymbol{\varepsilon}^p)^T \mathrm{d}\boldsymbol{\varepsilon}^p}}\mathrm{d}\boldsymbol{\varepsilon}^p$$

所以

$$\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}=\frac{\partial \bar{\varepsilon}^p}{\partial \boldsymbol{\varepsilon}^p}=\frac{\mathrm{d}\boldsymbol{\varepsilon}^p}{\sqrt{(\mathrm{d}\boldsymbol{\varepsilon}^p)^T \mathrm{d}\boldsymbol{\varepsilon}^p}}$$

将(5.30)式代入上式，得

$$\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}=\frac{\partial \bar{\varepsilon}^p}{\partial \boldsymbol{\varepsilon}^p}=\frac{(\partial f/\partial \boldsymbol{\sigma})}{\left[\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\frac{\partial f}{\partial \boldsymbol{\sigma}}\right]^{1/2}} \tag{5.39}$$

把（5.35）式代入（5.34）式得到加载时的本构方程，

$$\mathrm{d}\boldsymbol{\varepsilon}=\left[\boldsymbol{C}+\frac{1}{A}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\right]\mathrm{d}\boldsymbol{\sigma} \tag{5.40}$$

如果引入阶梯函数，

$$H(l)=\begin{cases}0 & (l\le 0)\\ 1 & (l>0)\end{cases} \tag{5.41}$$

则强化塑性材料的加载、中性变载和卸载普遍适用的增量型本构方程为

$$\mathrm{d}\boldsymbol{\varepsilon}=\left[\boldsymbol{C}+\frac{H(l)}{A}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\right]\mathrm{d}\boldsymbol{\sigma}=\boldsymbol{C}_{ep}\mathrm{d}\boldsymbol{\sigma} \tag{5.42}$$

根据上式，只要给出应力增量 $\mathrm{d}\boldsymbol{\sigma}$，就可以得到惟一确定的应变增量 $\mathrm{d}\boldsymbol{\varepsilon}$。

上面建立的本构方程是以应力矢量为基本变量，在应力空间中讨论的，仅适用于强化材料。对于理想弹塑性材料，由于 $A=0$，不能用一致性条件确定 $\mathrm{d}\lambda$ 的大小。对于软化材料不能用应力增量建立加-卸载准则。因而在应力空间中表述的本构理论有很大的局限性。

例 5-1 试讨论以下各向同性强化规律下的加载、卸载准则。

$$f=\frac{1}{2}\left[(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2+(\sigma_1-\sigma_2)^2\right]-\sigma_y^2=0$$

容易证明，上式也可以用应力偏量 $s_{ij}=\sigma_{ij}-\frac{1}{3}\sigma_{kk}\delta_{ij}$ 的第二不变量 J_2 来表示，即

$$f=\frac{1}{2}s_{ij}s_{ij}-\tau_y^2=J_2-\tau_y^2=0$$

上式中 τ_y^2 和 σ_y^2 存在一定关系，可参见 5.5 节的分析。

$$l=\frac{\partial f}{\partial\sigma_{kl}}\mathrm{d}\sigma_{kl}=\frac{\partial f}{\partial s_{mn}}\frac{\partial s_{mn}}{\partial\sigma_{kl}}\mathrm{d}\sigma_{kl}$$

σ_{kl} 是应力张量的分量，同时注意到，

$$\frac{\partial f}{\partial s_{mn}}=s_{mn}$$

$$\frac{\partial s_{mn}}{\partial\sigma_{kl}}=\frac{\partial(\sigma_{mn}-\frac{1}{3}\sigma_{pp}\delta_{mn})}{\partial\sigma_{kl}}=\delta_{mk}\delta_{nl}-\frac{1}{3}\delta_{kl}\delta_{mn}$$

因此有

$$\frac{\partial f}{\partial\sigma_{kl}}\mathrm{d}\sigma_{kl}=s_{mn}(\delta_{mk}\delta_{nl}-\frac{1}{3}\delta_{kl}\delta_{mn})\mathrm{d}\sigma_{kl}=s_{kl}\mathrm{d}\sigma_{kl}$$

上式推导过程中利用了应力偏量的第一不变量 $J_1=s_{kk}$ 等于零的性质。

$$s_{kl}\mathrm{d}\sigma_{kl}=s_{kl}\mathrm{d}(s_{kl}+\frac{1}{3}\sigma_{mm}\delta_{kl})=s_{kl}\mathrm{d}s_{kl}=\frac{1}{2}\mathrm{d}(s_{kl}s_{kl})=\mathrm{d}J_2$$

则

$$l=\mathrm{d}J_2$$

由此可以得到如下加卸载条件。

卸载：	$f=0$	$\mathrm{d}J_2<0$
中性变载：	$f=0$	$\mathrm{d}J_2=0$
加载：	$f=0$	$\mathrm{d}J_2>0$

5.4　应变空间表述的小变形弹塑性本构关系

由于应力空间描述的应力应变关系存在一定的缺陷，现在我们讨论如何在应变空间中表述弹塑性材料的本构性质。从图 5.4 的单轴应力的应力应变曲线上我们能看到，从自然状态出发，当 $\varepsilon<\varepsilon_y$ 时，介质的反应是线性弹性的，而当应变超过 ε_y 后（例如在 B 点）就伴有塑性变形发生，应力和应变之间的关系不再是线性的了。在卸载之后重新加载时，在应变重新达到 B 点对应的值 ε_{yB} 之前，反应是线性弹性的，而达到这个应变值时，才可能产生新的塑性应变。这样，ε_{yB} 可以看作是用应变表示的后继屈服极限。在塑性状态的 B 点，可用 $\mathrm{d}\varepsilon>0$ 表示塑性加载，用 $\mathrm{d}\varepsilon<0$ 表示塑性卸载。而且这种表示在软化阶段也是正确的，例如在点 C 表示的塑性状态，卸载时显然有 $\mathrm{d}\varepsilon<0$，在加载时虽然有 $\mathrm{d}\varepsilon^e<0$（即 $\mathrm{d}\sigma=E\mathrm{d}\varepsilon^e<0$），但由于 $\mathrm{d}\varepsilon^p>0$，仍有 $\mathrm{d}\varepsilon>0$。这样就可期望在应变空间中能给出对强化、软化和理想塑性普遍适用的本构理论

表述。

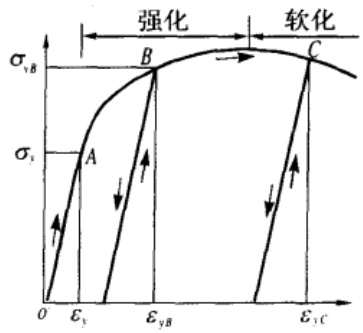


图 5.4 弹塑性材料

5.4.1 应变空间中的屈服条件与屈服面

将前面阐述的简单应力状态下的应变屈服极限推广到一般的应变状态，就得到应变屈服条件和应变屈服面。用六维应变空间中一个点代表介质的应变状态。假设存在一张超曲面，它所包围的区域内部所有的点都能用纯弹性应变的路径达到，而在这个超曲面上的点表示介质将会发生进一步的塑性变形，这个曲面就是应变空间的屈服面，它可表示为

$$F(\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa)=0 \tag{5.43}$$

其中应变矢量 $\boldsymbol{\varepsilon}$ 是基本变量，塑性应变矢量 $\boldsymbol{\varepsilon}^p$ 和标量 κ 是内变量。这样，(5.43) 式表示以内变量为参数的一族超曲面。如果 (5.43) 中 $\boldsymbol{\varepsilon}^p=\mathbf{0},\kappa=0$，就得到初始屈服面，

$$F^*(\boldsymbol{\varepsilon})=0 \tag{5.44}$$

在应变空间中，介质质点的状态可用 $\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa$ 表示。如果有

$$F(\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa)<0$$

则这个状态处于弹性状态，对无限小的外部作用的反应是纯弹性的。如果在这个状态有

$$F(\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa)=0$$

那么它是塑性状态，对无限小的外部作用的反应是弹塑性的。不存在使 $F>0$ 的状态。

5.4.2 应变空间中的加载、卸载准则

设某质点 $\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa$ 处于塑性状态，这时 $F(\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}^p,\kappa)=0$，即应变点在屈服面上。在外部作用下，如果应变点保持在屈服面上，并有新的塑性变形发生，这个过程叫做塑性加载（简称加载）；如果应变点离开屈服面，退回弹性区，反应是纯弹性的，叫做塑性卸载（简称卸载）。它们的一个中间情况，即应变点不离开屈服面，而又无新的塑性变形发生，为中性变载。可以证明，在应变空间中表述的加-卸载准则是，

$$F=0 \qquad L=\left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T \mathrm{d}\boldsymbol{\varepsilon} \begin{cases} <0 & \text{卸载} \\ =0 & \text{中性变载} \\ >0 & \text{加载} \end{cases} \tag{5.45}$$

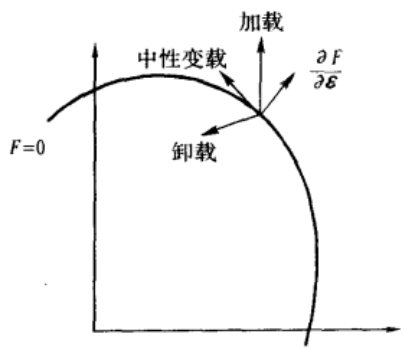


图5.5 应变空间中的加、卸载准则

这个准则对强化、软化和理想塑性材料是普遍适用的。这个准则的几何解释如图 5.5 所示。对于塑性状态，加载、卸载和中性变载分别对应于应变增量矢量 $d\boldsymbol{\varepsilon}$ 指向应变屈服面的外侧、内侧和与屈服面相切。

5.4.3　应变空间中的关联流动法则和一致性条件

理论上同样可以证明，对于弹性性质不随塑性变形的出现和发展而变化的材料，加载时塑性应力增量 $d\boldsymbol{\sigma}^p$ 指向应变空间中屈服面的外法向，即

$$d\boldsymbol{\sigma}^p = d\lambda \frac{\partial F}{\partial \boldsymbol{\varepsilon}} \tag{5.46}$$

其中，$d\lambda$ 是一个非负的尺度因子，加载时 $d\lambda > 0$，其他情况 $d\lambda = 0$。（5.46）式表述的是应变空间中的关联流动法则，也称为在应变空间中表述的正交法则。

在加载过程中，应变状态始终处于屈服面上，即在加载过程中，F=0 始终成立。则有

$$dF = \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T d\boldsymbol{\varepsilon} + \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}^p}\right)^T d\boldsymbol{\varepsilon}^p + \frac{\partial F}{\partial \kappa} d\kappa = 0 \tag{5.47}$$

上式即为弹塑性材料塑性加载时的一致性条件。同应力空间表示的一致性条件相似，式(5.47)表征加载过程中，微小应变增量和微小塑性应变增量间必须满足的关系。

5.4.4　应变空间表述的应力应变关系

类似地，首先假设无限小应变增量 $d\boldsymbol{\varepsilon}$ 可分解为弹性部分与塑性部分之和

$$d\boldsymbol{\varepsilon} = d\boldsymbol{\varepsilon}^e + d\boldsymbol{\varepsilon}^p \tag{5.48}$$

而其中的弹性部分 $d\boldsymbol{\varepsilon}^e$ 与应力增量 $d\boldsymbol{\sigma}$ 之间满足胡克定律，

$$d\boldsymbol{\sigma} = \boldsymbol{D} d\boldsymbol{\varepsilon}^e \tag{5.49}$$

考虑到流动法则（5.46）式，由（5.48）和（5.49）式得

$$d\boldsymbol{\sigma} = \boldsymbol{D} d\boldsymbol{\varepsilon}^e = \boldsymbol{D}(d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p) = \boldsymbol{D} d\boldsymbol{\varepsilon} - d\boldsymbol{\sigma}^p = \boldsymbol{D} d\boldsymbol{\varepsilon} - d\lambda \frac{\partial F}{\partial \boldsymbol{\varepsilon}} \tag{5.50}$$

在中性变载和卸载时，$d\lambda = 0$，上式退化为胡克定律。对于加载，$d\lambda > 0$，$d\lambda$ 的大小由一致性条件（5.47）式来确定。具体推导如下，

$$\begin{aligned}
dF &= \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T d\boldsymbol{\varepsilon} + \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}^p}\right)^T d\boldsymbol{\varepsilon}^p + \frac{\partial F}{\partial \kappa} d\kappa \\
&= \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T d\boldsymbol{\varepsilon} + \left[\left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}^p}\right)^T + \frac{\partial F}{\partial \kappa}\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\right] d\boldsymbol{\varepsilon}^p \\
&= \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T d\boldsymbol{\varepsilon} + \left[\left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}^p}\right)^T + \frac{\partial F}{\partial \kappa}\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\right] \boldsymbol{C} d\boldsymbol{\sigma}^p \\
&= \left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}}\right)^T d\boldsymbol{\varepsilon} + \left[\left(\frac{\partial F}{\partial \boldsymbol{\varepsilon}^p}\right)^T + \frac{\partial F}{\partial \kappa}\left(\frac{\partial \kappa}{\partial \boldsymbol{\varepsilon}^p}\right)^T\right] \boldsymbol{C} \cdot \frac{\partial F}{\partial \boldsymbol{\varepsilon}} \cdot d\lambda = 0
\end{aligned}$$

由上式可求得

$$d\lambda = \frac{1}{A}\left(\frac{\partial F}{\partial \varepsilon}\right)^T d\varepsilon \tag{5.51}$$

其中，

$$A = -\left(\frac{\partial F}{\partial \varepsilon^p}\right)^T \boldsymbol{C} \frac{\partial F}{\partial \varepsilon} - M \tag{5.52}$$

$$M = \begin{cases} \dfrac{\partial F}{\partial \kappa} \boldsymbol{\sigma}^T \boldsymbol{C} \dfrac{\partial F}{\partial \varepsilon} & \text{当} \kappa = \omega^p \\ \dfrac{\partial F}{\partial \kappa} \boldsymbol{m}^T \boldsymbol{C} \dfrac{\partial F}{\partial \varepsilon} & \text{当} \kappa = \theta^p \\ \dfrac{\partial F}{\partial \kappa}\left(\left(\dfrac{\partial F}{\partial \kappa}\right)^T \boldsymbol{C}\boldsymbol{C} \dfrac{\partial F}{\partial \varepsilon}\right)^{1/2} & \text{当} \kappa = \bar{\varepsilon}^p \end{cases} \tag{5.53}$$

将（5.51）式代入（5.50）式得加载时的本构方程，

$$d\boldsymbol{\sigma} = (\boldsymbol{D} - \boldsymbol{D}_p) d\boldsymbol{\varepsilon} \tag{5.54}$$

$$\boldsymbol{D}_p = \frac{1}{A} \frac{\partial F}{\partial \varepsilon}\left(\frac{\partial F}{\partial \varepsilon}\right)^T \tag{5.55}$$

利用阶梯函数，可以给出对加载、卸载和中性变载普遍适用的本构方程形式，

$$d\boldsymbol{\sigma} = (\boldsymbol{D} - H(L)\boldsymbol{D}_p) d\boldsymbol{\varepsilon} \equiv \boldsymbol{D}_{ep} d\boldsymbol{\varepsilon} \tag{5.56}$$

其中，$\boldsymbol{D}_p$ 称为塑性矩阵，$\boldsymbol{D}_{ep}$ 称为弹塑性矩阵，且

$$L = \left(\frac{\partial F}{\partial \varepsilon}\right)^T d\varepsilon$$

它是应变空间中的加-卸载准则函数。

比较(5.42)式和（5.56）式可以看出，使用本构方程（5.56），可由应变增量 $d\boldsymbol{\varepsilon}$ 惟一地确定应力增量 $d\boldsymbol{\sigma}$，这与位移法有限元分析过程一致。

5.4.5 应变空间描述与应力空间描述的关系

由于对屈服函数的实验研究多是在控制载荷的试验机上进行的，因此屈服条件往往是用应力分量表示的。塑性力学的许多基本概念，例如强化、软化和理想塑性等，也都是用应力分量定义的，关于应变空间的屈服面的知识却很少。因此，必须建立起应变空间描述和应力空间描述的弹塑性本构关系间的联系。首先建立应力屈服函数和应变屈服函数间的关系。

$$f(\boldsymbol{\sigma}, \boldsymbol{\sigma}^p, \kappa) = f(\boldsymbol{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p), \boldsymbol{D}\boldsymbol{\varepsilon}^p, \kappa) \equiv F(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^p, \kappa) \tag{5.57}$$

由此不难得出自变量偏导数间的关系，

$$\frac{\partial F}{\partial \varepsilon} = \boldsymbol{D}\frac{\partial f}{\partial \sigma} \qquad \frac{\partial F}{\partial \varepsilon^p} = \boldsymbol{D}\left(\frac{\partial f}{\partial \sigma^p} - \frac{\partial f}{\partial \sigma}\right) \qquad \frac{\partial F}{\partial \kappa} = \frac{\partial f}{\partial \kappa} \tag{5.58}$$

将(5.58)式代入(5.55)式得

$$\boldsymbol{D}_p = \frac{1}{A}\frac{\partial F}{\partial \varepsilon}\left(\frac{\partial F}{\partial \varepsilon}\right)^T = \frac{1}{A}\boldsymbol{D}\frac{\partial f}{\partial \sigma}\left(\frac{\partial f}{\partial \sigma}\right)^T \boldsymbol{D} \tag{5.59}$$

将(5.58)式代入(5.52)式，得

$$A=-\left(\frac{\partial F}{\partial \varepsilon^{p}}\right)^{T} \boldsymbol{C} \frac{\partial F}{\partial \varepsilon}-M=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \boldsymbol{D} \frac{\partial f}{\partial \boldsymbol{\sigma}}-\left(\frac{\partial f}{\partial \boldsymbol{\sigma}^{p}}\right)^{T} \boldsymbol{D} \frac{\partial f}{\partial \boldsymbol{\sigma}}-M \tag{5.60}$$

将(5.58)式代入(5.53)式，得

$$M=\begin{cases}\dfrac{\partial f}{\partial \kappa} \boldsymbol{\sigma}^{T} \dfrac{\partial f}{\partial \boldsymbol{\sigma}} & \text{当} \kappa=w^{p} \\ \dfrac{\partial f}{\partial \kappa} \boldsymbol{M}^{T} \dfrac{\partial f}{\partial \boldsymbol{\sigma}} & \text{当} \kappa=\theta^{p} \\ \dfrac{\partial f}{\partial \kappa}\left[\left(\dfrac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \dfrac{\partial f}{\partial \boldsymbol{\sigma}}\right]^{1/2} & \text{当} \kappa=\bar{\varepsilon}^{p}\end{cases} \tag{5.61}$$

将(5.58)式代入(5.45)式，得

$$L=\left(\frac{\partial F}{\partial \varepsilon}\right)^{T} \mathrm{d}\varepsilon=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \boldsymbol{D} \mathrm{d}\varepsilon=\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \mathrm{d}\boldsymbol{\sigma}^{e} \tag{5.62}$$

式中，$\mathrm{d}\boldsymbol{\sigma}^{e}=\boldsymbol{D}\mathrm{d}\boldsymbol{\varepsilon}$ 是由 $\mathrm{d}\boldsymbol{\varepsilon}$ 按弹性规律计算的“弹性”应力增量，其性质类似 $\mathrm{d}\boldsymbol{\sigma}^{p}$，也不是客观存在的。比较式（5.62）和（5.29）可以看出，应力空间描述的加卸载准则函数 l 和用应力空间的屈服函数表示的、应变空间描述的加卸载准则函数 L 在形式上是相似的，但本质上是不同的。两者的区别在于用的是应力增量还是“弹性”应力增量。同时，在位移法的有限元分析中，我们只能用应变空间表述的准则，因为在加卸载判断时，$\mathrm{d}\boldsymbol{\sigma}$ 还是待求的，事先并不知道。

至此为止，我们已经给出了弹塑性介质的完整的小变形本构关系，在一个已知的状态 $\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^{p}, \kappa$（或 $\boldsymbol{\sigma}, \boldsymbol{\sigma}^{p}, \kappa$）的基础上，由应变增量 $\mathrm{d}\boldsymbol{\varepsilon}$ 可以惟一确定应力增量 $\mathrm{d}\boldsymbol{\sigma}$，具体的步骤为：

（1）由塑性应变 $\boldsymbol{\varepsilon}^{p}$（或 $\boldsymbol{\sigma}^{p}$）和内时变量 κ 确定屈服函数 F（或 f）。

（2）如果应变 $\boldsymbol{\varepsilon}$（或应力 $\boldsymbol{\sigma}$）使 $F<0$（或 $f<0$），本构方程是

$$\mathrm{d}\boldsymbol{\sigma}=\boldsymbol{D}\mathrm{d}\boldsymbol{\varepsilon}$$

反应是纯弹性的。

（3）如果 $\boldsymbol{\varepsilon}$（或 $\boldsymbol{\sigma}$）使 $F=0$（或 $f=0$），反应是弹塑性的，可根据加载、卸载准则进行判断。

(i) 当 $\left(\dfrac{\partial F}{\partial \varepsilon}\right)^{T} \mathrm{d}\varepsilon=\left(\dfrac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \mathrm{d}\boldsymbol{\sigma}^{e} \leq 0$ 时为中性变载或卸载，反应为纯弹性的，$\mathrm{d}\boldsymbol{\sigma}=\boldsymbol{D}\mathrm{d}\boldsymbol{\varepsilon}$。

(ii) 当 $\left(\dfrac{\partial F}{\partial \varepsilon}\right)^{T} \mathrm{d}\varepsilon=\left(\dfrac{\partial f}{\partial \boldsymbol{\sigma}}\right)^{T} \mathrm{d}\boldsymbol{\sigma}^{e}>0$ 时为加载，反应为弹塑性的，$\mathrm{d}\boldsymbol{\sigma}=(\boldsymbol{D}-\boldsymbol{D}_{p})\mathrm{d}\boldsymbol{\varepsilon}$。

5.5　几种常见的弹塑性模型

本节介绍几种常见的弹塑性模型。

5.5.1　等向强化-软化的米塞斯（Mises）材料

Mises 等向强化-软化材料的屈服条件是，

$$f = J_2^{1/2} - \mathrm{K}(\kappa) = 0 \tag{5.63}$$

K 为内时变量 κ 的函数，一般可以由简单应力状态下的试验数据给出。

一、纯剪切试验

纯剪切状态下，假设除 $\tau_{12} = \tau_{21} = \tau$ 非零外，其他应力分量均为零，则应力张量分量可表示为

$$\sigma_{ij} = \tau(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1})$$

偏应力张量分量为

$$s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij} = \tau(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1})$$

偏应力张量第二不变量为

$$J_2 = \frac{1}{2}s_{ij}s_{ij} = \frac{1}{2}\tau^2(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1})(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1}) = \tau^2$$

代入(5.63)式有

$$f = \tau - \mathrm{K}(\kappa) = 0$$

因此，$\mathrm{K}(\kappa)$ 为剪切屈服应力 τ_y，即 $\mathrm{K}(\kappa) = \tau_y(\kappa)$。

二、简单拉伸试验

简单拉伸状态下，假设除 $\sigma_{11} = \sigma$ 非零外，其他应力分量均为零，则应力张量分量可表示为

$$\sigma_{ij} = \sigma\delta_{i1}\delta_{j1}$$

应力偏量的分量为

$$s_{ij} = \sigma\delta_{i1}\delta_{j1} - \frac{1}{3}\sigma\delta_{k1}\delta_{k1}\cdot\delta_{ij} = \sigma\delta_{i1}\delta_{j1} - \frac{1}{3}\sigma\delta_{ij} = \sigma(\delta_{i1}\delta_{j1} - \frac{1}{3}\delta_{ij})$$

偏应力张量第二不变量为

$$J_2 = \frac{1}{2}s_{ij}s_{ij} = \frac{1}{2}[\delta_{i1}\delta_{j1} - \frac{1}{3}\delta_{ij}][\delta_{i1}\delta_{j1} - \frac{1}{3}\delta_{ij}]\sigma^2 = \frac{1}{3}\sigma^2$$

代入(5.63)式有

$$f = \frac{1}{\sqrt{3}}\sigma - \mathrm{K}(\kappa) = 0$$

因此有 $\mathrm{K}(\kappa) = \frac{1}{\sqrt{3}}\sigma_y$。

显然，如果材料满足 Mises 屈服条件，应当有 $\sigma_y = \sqrt{3}\tau_y$。由于屈服条件(5.63)和平均应力无关，在主应力空间中，Mises 屈服面为垂直于 π 平面 $(\sigma_1 + \sigma_2 + \sigma_3 = 0)$ 的圆柱面（见图 5.6）。

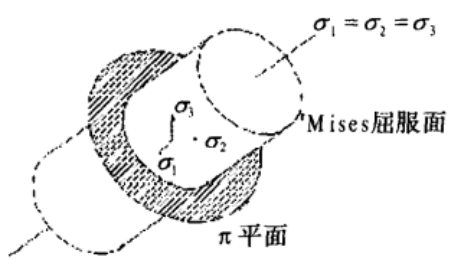


图5.6 Mises屈服面

下面推导 Mises 材料的弹塑性应力应变关系。

一、空间问题

为方便起见，先求偏应力张量第二不变量关于应力矢量分量的偏导数。请注意，推导过程中作为应力矢量的分量，σ_{ij} 和 σ_{ji} 是同一个量，而作为应力张量的分量，它们是各自独立的

两个物理量。因此，

$$\frac{\partial J_2}{\partial \sigma_{\overline{kk}}} = s_{11}\frac{\partial s_{11}}{\partial \sigma_{\overline{kk}}} + s_{22}\frac{\partial s_{22}}{\partial \sigma_{\overline{kk}}} + s_{33}\frac{\partial s_{33}}{\partial \sigma_{\overline{kk}}} = s_{\overline{kk}}$$

$$\frac{\partial J_2}{\partial \sigma_{ij}} = 2s_{kl}\frac{\partial s_{kl}}{\partial \sigma_{ij}} = 2s_{ij} \quad (i \neq j)$$

写成矢量形式，

$$\frac{\partial J_2}{\partial \boldsymbol{\sigma}} = \{s_{11}, s_{22}, s_{33}, 2s_{23}, 2s_{31}, 2s_{12}\}^T \overset{\Delta}{=} \overline{\boldsymbol{s}}$$

不难求得

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{1}{2\sqrt{J_2}}\frac{\partial J_2}{\partial \boldsymbol{\sigma}} = \frac{1}{2\sqrt{J_2}}\overline{\boldsymbol{s}} = \frac{1}{2\mathrm{K}(\kappa)}\overline{\boldsymbol{s}} \tag{5.64}$$

$$\boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{1}{2\mathrm{K}(\kappa)}\boldsymbol{D}\,\overline{\boldsymbol{s}}$$

对于各向同性材料，

$$\boldsymbol{D} = \begin{bmatrix} K+4G/3 & K-2G/3 & K-2G/3 & & & \\ & K+4G/3 & K-2G/3 & & & \\ & & K+4G/3 & & & \\ & \text{对称} & & G & & \\ & & & & G & \\ & & & & & G \end{bmatrix}$$

容易证明，

$$\boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{G}{\mathrm{K}(\kappa)}\boldsymbol{s} \tag{5.65}$$

式中，

$$\boldsymbol{s} = \{s_{11}, s_{22}, s_{33}, s_{23}, s_{31}, s_{12}\}^T$$

利用 $J_2 = \frac{1}{2}\overline{\boldsymbol{s}}^T\boldsymbol{s} = \frac{1}{2}\boldsymbol{s}^T\overline{\boldsymbol{s}}$ 可得

$$\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{G}{\mathrm{K}(\kappa)}\cdot\frac{1}{2\mathrm{K}(\kappa)}\overline{\boldsymbol{s}}^T\boldsymbol{s} = G \tag{5.66}$$

如果取 $\kappa = w^p$，则

$$-M = -\frac{\partial f}{\partial w^p}\boldsymbol{\sigma}^T\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{\mathrm{dK}(w^p)}{\mathrm{d}w^p}\boldsymbol{\sigma}^T\frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{\mathrm{dK}(w^p)}{\mathrm{d}w^p}\boldsymbol{\sigma}^T\frac{1}{2\mathrm{K}(w^p)}\overline{\boldsymbol{s}} = \frac{\mathrm{d}K(w^p)}{\mathrm{d}w^p}\cdot\frac{1}{2\mathrm{K}(w^p)}\boldsymbol{\sigma}^T\overline{\boldsymbol{s}}$$

注意到，$\boldsymbol{\sigma}^T\overline{\boldsymbol{s}} = \boldsymbol{s}^T\overline{\boldsymbol{s}} = 2J_2 = 2\mathrm{K}^2(w^p)$，则

$$-M = \mathrm{K}(w^p)\cdot\frac{\mathrm{dK}(w^p)}{\mathrm{d}w^p} = \frac{1}{2}\frac{\mathrm{dK}^2(w^p)}{\mathrm{d}w^p} \tag{5.67}$$

在纯剪切情况下，$\mathrm{K}(w^p) = \tau_y(w^p), w^p = \int \tau_y \mathrm{d}\gamma^p$，则

$$\mathrm{dK}^2(w^p) = 2\tau_y \mathrm{d}\tau_y$$

$$dw^p = \tau_y d\gamma^p$$

将上述两式代入(5.67)式，得

$$-M = \frac{d\tau_y}{d\gamma^p} = G_p \tag{5.68}$$

显然，G_p 为 $\tau_s - \gamma^p$ 曲线的斜率，称为剪切塑性模量。

在单向拉伸情况下，$K(w^p) = \sigma_y / \sqrt{3}, w^p = \int \sigma_y d\varepsilon^p$，则

$$dK^2(w^p) = \frac{1}{3}\sigma_y d\sigma_y$$

$$dw^p = \sigma_y d\varepsilon^p$$

此时，

$$-M = \frac{1}{3}\frac{d\sigma_y}{d\varepsilon^p} = \frac{1}{3}E_p \tag{5.69}$$

同样，E_p 是 $\sigma_y - \varepsilon^p$ 曲线的斜率，称为拉伸塑性模量。将(5.66)、(5.68)和(5.69)式代入(5.60)式，并注意到 $\frac{\partial f}{\partial \boldsymbol{\sigma}^p} = \mathbf{0}$，有

$$\begin{aligned} A &= \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}} - \left(\frac{\partial f}{\partial \boldsymbol{\sigma}^p}\right)^T \boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}} - M = G + \frac{1}{2}\frac{dK^2(w^p)}{dw^p} \\ &= G + G_p(w^p) = G + \frac{1}{3}E_p(w^p) \end{aligned} \tag{5.70}$$

利用(5.65)式可得

$$\boldsymbol{D}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \boldsymbol{D} = \frac{G^2}{K^2(\kappa)}\boldsymbol{s}\cdot\boldsymbol{s}^T \tag{5.71}$$

把（5.70）和（5.71）式代入（5.59）式，可得

$$\boldsymbol{D}_p = \frac{G}{(1 + G_p/G)K^2}\boldsymbol{s}\,\boldsymbol{s}^T \tag{5.72}$$

显然，当 $G_p > 0$ 时表示材料强化屈服，$G_p = 0$ 表示材料为理想弹塑性，当 $G_p < 0$ 表示材料软化屈服。容易证明，Mises 材料是塑性体积不可压缩的。

二、轴对称问题和平面应变问题

轴对称问题和平面应变问题可按同一格式处理。它们都只有四个非零的应力、应变分量。

轴对称问题：

$$\boldsymbol{\sigma} = \{\sigma_r,\ \sigma_z,\ \sigma_\theta,\ \tau_{rz}\}^T$$

$$\boldsymbol{\varepsilon} = \{\varepsilon_r,\ \varepsilon_z,\ \varepsilon_\theta,\ \gamma_{rz}\}^T$$

平面应变问题：

$$\boldsymbol{\sigma} = \{\sigma_x,\ \sigma_y,\ \sigma_z,\ \tau_{xy}\}^T$$

$$\boldsymbol{\varepsilon} = \{\varepsilon_x,\ \varepsilon_y,\ \varepsilon_z,\ \gamma_{xy}\}^T$$

所以只要在三维空间应力应变关系中划去二行二列，保留相应的四行四列，就可以得到平面

应变问题和轴对称问题的 $\boldsymbol{D}$，$\boldsymbol{D}_p$ 和 $\boldsymbol{D}_{ep}$。

三、平面应力问题

不失一般性，假设 $\sigma_{33}=\tau_{13}=\tau_{23}=0$。为引入 $\sigma_{33}=\tau_{13}=\tau_{23}=0$ 的条件，利用在 σ_{11}、σ_{22} 和 σ_{12} 构成的应力子空间中的屈服函数和应力应变关系比较方便。这时，应力矢量、应变矢量和各向同性弹性矩阵为

$$\boldsymbol{\sigma}=\{\sigma_{11},\ \sigma_{22},\ \tau_{12}\}^T$$

$$\boldsymbol{\varepsilon}=\{\varepsilon_{11},\ \varepsilon_{22},,\ \gamma_{12}\}^T$$

$$\boldsymbol{D}=\frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & \\ \nu & 1 & \\ & & (1-\nu)/2\end{bmatrix}$$

此时，应力偏量分量为

$$s_{11}=\frac{1}{3}(2\sigma_{11}-\sigma_{22})\qquad s_{22}=\frac{1}{3}(2\sigma_{22}-\sigma_{11})\qquad s_{33}=-\frac{1}{3}(\sigma_{11}+\sigma_{22})$$

$$s_{12}=\sigma_{12}\quad s_{23}=s_{31}=0$$

应力子空间中的屈服条件为

$$J_2=\frac{1}{3}(\sigma_{11}^2+\sigma_{22}^2-\sigma_{11}\sigma_{22}+3\sigma_{12}^2)$$

利用类似的推导方法可得

$$\frac{\partial J_2}{\partial\boldsymbol{\sigma}}=\{s_{11},s_{22},2s_{12}\}^T$$

将以上各式代入(5.59)~(5.61)式，得

$$\boldsymbol{D}_p=\frac{3E^2}{4(1-\nu^2)^2A\sigma_y^2}\begin{Bmatrix}s_{11}+\nu s_{22}\\ \nu s_{11}+s_{22}\\ (1-\nu)s_{12}\end{Bmatrix}\{s_{11}+\nu s_{22}\quad \nu s_{11}+s_{22}\quad (1-\nu)s_{12}\}$$

$$A=\frac{3E}{4(1-\nu^2)\sigma_y^2}\left(s_{11}^2+s_{22}^2+2\nu s_{11}s_{22}+2(1-\nu)s_{12}^2\right)-M$$

$$-M=\frac{1}{2}\frac{\mathrm{dK}^2(w^p)}{\mathrm{d}w^p}=\frac{1}{3}E_p$$

将以上两式代入 $\boldsymbol{D}_p$，并展开为矩阵形式，

$$\boldsymbol{D}_p=\frac{E}{B(1-\nu^2)}\begin{bmatrix}(s_{11}+\nu s_{22})^2 & (s_{11}+\nu s_{22})(s_{22}+\nu s_{11}) & (1-\nu)(s_{11}+\nu s_{22})s_{12}\\ (s_{11}+\nu s_{22})(s_{22}+\nu s_{11}) & (s_{22}+\nu s_{11})^2 & (1-\nu)(s_{22}+\nu s_{11})s_{12}\\ (1-\nu)(s_{11}+\nu s_{22})s_{12} & (1-\nu)(s_{22}+\nu s_{11})s_{12} & (1-\nu)^2s_{12}^2\end{bmatrix}\tag{5.73}$$

式中，

$$B=s_{11}^2+s_{22}^2+2\nu s_{11}s_{22}+2(1-\nu)s_{12}^2+\frac{4(1-\nu^2)E_p\sigma_y^2}{9E}\tag{5.74}$$

由此，不难得到平面应力问题的弹塑性矩阵。

5.5.2 随动强化的 Mises 材料

随动强化的 Mises 材料的屈服条件是，

$$f=\left[\frac{1}{2}(s_{ij}-\alpha\sigma_{ij}^{p})(s_{ij}-\alpha\sigma_{ij}^{p})\right]^{1/2}-\mathrm{K}_0=0 \tag{5.75}$$

式中，α 为与内时变量 κ 有关的标量，K_0 是常数，表示屈服面的大小不发生变化。而 $\sigma_{ij}^{p}=\int \mathrm{d}\sigma_{ij}^{p}$。类似于应力矢量 $\boldsymbol{\sigma}$，引入矢量记号

$$\boldsymbol{\sigma}^{p}=\{\sigma_{11}^{p},\sigma_{22}^{p},\sigma_{33}^{p},\sigma_{23}^{p},\sigma_{31}^{p},\sigma_{12}^{p}\}^{T}$$

它确定了屈服面在应力空间中的位置，称为应力迁移矢量，或背应力矢量。对于各向同性材料，由（5.75）式可以导出，

$$\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{1}{2\mathrm{K}_0}(\bar{\boldsymbol{s}}-\alpha\bar{\boldsymbol{\sigma}}^{p}) \tag{5.76}$$

$$\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{1}{2\mathrm{K}_0}\boldsymbol{D}(\bar{\boldsymbol{s}}-\alpha\bar{\boldsymbol{\sigma}}^{p})=\frac{G}{\mathrm{K}_0}(\boldsymbol{s}-\alpha\boldsymbol{\sigma}^{p}) \tag{5.77}$$

式中，

$$\bar{\boldsymbol{\sigma}}^{p}=\{\sigma_{11}^{p},\sigma_{22}^{p},\sigma_{33}^{p},2\sigma_{23}^{p},2\sigma_{31}^{p},2\sigma_{12}^{p}\}^{T}$$

容易求得

$$\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{G}{2\mathrm{K}_0^{2}}(\bar{\boldsymbol{s}}-\alpha\bar{\boldsymbol{\sigma}}^{p})^{T}(\boldsymbol{s}-\alpha\boldsymbol{\sigma}^{p})=G \tag{5.78}$$

$$-\left(\frac{\partial f}{\partial\boldsymbol{\sigma}^{p}}\right)^{T}\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{\alpha}{2\mathrm{K}_0}(\bar{\boldsymbol{s}}-\alpha\bar{\boldsymbol{\sigma}}^{p})^{T}\frac{G}{\mathrm{K}_0}(\boldsymbol{s}-\alpha\boldsymbol{\sigma}^{p})=\alpha G \tag{5.79}$$

$$M=0 \tag{5.80}$$

将以上各式代入（5.59）式可得随动强化 Mises 材料的塑性矩阵，

$$\boldsymbol{D}_{p}=\frac{G}{(1+\alpha)\mathrm{K}_0^{2}}(\boldsymbol{s}-\alpha\boldsymbol{\sigma}^{p})(\boldsymbol{s}-\alpha\boldsymbol{\sigma}^{p})^{T} \tag{5.81}$$

比较（5.72）和（5.81）式可见，两者在形式上是相同的。轴对称问题、平面应变问题和平面应力问题的弹塑性矩阵可类似推导，这里不再重复。

5.5.3 等向强化-软化的 Drucker-Prager 材料

等向强化-软化的 Drucker-Prager 材料的屈服条件是

$$f=J_2^{1/2}+\alpha I_1-\mathrm{K}=0 \tag{5.82}$$

式中，I_1 为应力张量的第一不变量，即

$$I_1=\sigma_{kk}=\sigma_{11}+\sigma_{22}+\sigma_{33} \tag{5.83}$$

式中，α 和 K 为材料系数，一般随塑性应变的发展而改变。显然，屈服面（5.82）在主应力空间中是一个圆锥面，其在 π 平面上为一个圆。与 Mises 等向强化和随动强化材料不同，Drucker-Prager 屈服条件考虑了围压（即静水压力）对屈服面的影响，并且能够反映剪切引起膨胀（扩容）的性质，适合模拟岩土、混凝土一类材料的弹塑性变形。因此，这种模型在土

木工程中得到了广泛的应用。

为确定参数 α 和 K 与土木工程中常用的内聚力 c 和摩擦角 ϕ 的关系，需要将 Drucker-Prager 屈服条件与 Coulomb 屈服条件作比较。

$$f(\sigma_1,\sigma_2,\sigma_3)=\frac{1}{2}(\sigma_1-\sigma_3)+\frac{1}{2}(\sigma_1+\sigma_3)\sin\phi-c\cos\phi=0 \tag{5.84}$$

式中，假设 $\sigma_1\geq\sigma_2\geq\sigma_3$，若无此规定，需采用对称开拓法修正上述屈服条件，同时注意到正应力以拉为正，这与岩土工程中应力以压为正的定义是不同的。容易证明，Coulomb 屈服条件在主应力空间中是一个六棱锥，其在 π 平面中的截线是一个不规则的六边形，称为 Coulomb 六边形（如图 5.7 所示）。使 Drucker-Prager 屈服面的锥顶和 Coulomb 棱锥的锥顶重合。当 Drucker-Prager 屈服面在 π 平面上的圆与 Coulomb 六边形的外顶点重合时，可得

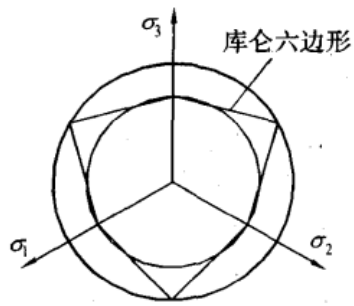


图 5.7　Prucker-Prager 图

$$\alpha=\frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)}\qquad \mathrm{K}=\frac{6c\cos\phi}{\sqrt{3}(3-\sin\phi)} \tag{5.85}$$

当与 Coulomb 六边形的内顶点重合时，可得

$$\alpha=\frac{2\sin\phi}{\sqrt{3}(3+\sin\phi)}\qquad \mathrm{K}=\frac{6c\cos\phi}{\sqrt{3}(3+\sin\phi)} \tag{5.86}$$

在平面应变情况下，可得

$$\alpha=\frac{\tan\phi}{(9+12\tan^2\phi)^{1/2}}\qquad \mathrm{K}=\frac{3c}{(9+12\tan^2\phi)^{1/2}} \tag{5.87}$$

对于各向同性材料，由（5.82）式不难得出，

$$\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{1}{2\sqrt{J_2}}\bar{\boldsymbol{s}}+\alpha\boldsymbol{m} \tag{5.88}$$

$$\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{G}{\sqrt{J_2}}\boldsymbol{s}+3\alpha K\boldsymbol{m} \tag{5.89}$$

$$\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\left(\frac{1}{2\sqrt{J_2}}\bar{\boldsymbol{s}}+\alpha\boldsymbol{m}\right)^T\left(\frac{G}{\sqrt{J_2}}\boldsymbol{s}+3\alpha K\boldsymbol{m}\right)=9\alpha^2K+G \tag{5.90}$$

如果取 $\kappa=\theta^p$，$\mathrm{K}=\mathrm{K}(\theta^p)$，$\alpha=\alpha(\theta^p)$，则

$$-M=-\frac{\partial f}{\partial\theta^p}\boldsymbol{m}^T\frac{\partial f}{\partial\boldsymbol{\sigma}}=\left(\frac{\partial\mathrm{K}}{\partial\theta^p}-\frac{\partial\alpha}{\partial\theta^p}I_1\right)\boldsymbol{m}^T\left(\alpha\boldsymbol{m}+\frac{1}{2\sqrt{J_2}}\bar{\boldsymbol{s}}\right)=\left(\frac{\partial\mathrm{K}}{\partial\theta^p}-\frac{\partial\alpha}{\partial\theta^p}I_1\right)3\alpha \tag{5.91}$$

$$A=9\alpha^2K+G+3\alpha\left(\frac{\partial\mathrm{K}}{\partial\theta^p}-\frac{\partial\alpha}{\partial\theta^p}I_1\right) \tag{5.92}$$

$$\frac{1}{A^{1/2}}\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{1}{A^{1/2}}\left[\frac{G}{\sqrt{J_2}}\boldsymbol{s}+3\alpha K\boldsymbol{m}\right]=\beta_1\boldsymbol{s}+\beta_2\boldsymbol{m} \tag{5.93}$$

式中，

$$\beta_1=\frac{G}{A^{1/2}J_2^{1/2}} \qquad \beta_2=\frac{3\alpha K}{A^{1/2}}$$

代入塑性矩阵的表达式得

$$\boldsymbol{D}_p=\left[\frac{1}{A^{1/2}}\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}\right]\left[\frac{1}{A^{1/2}}\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}\right]^T=[\beta_1\boldsymbol{S}+\beta_2\boldsymbol{m}][\beta_1\boldsymbol{S}^T+\beta_2\boldsymbol{m}^T] \tag{5.94}$$

5.5.4 层状材料

节理和裂隙有一定取向的节理岩石以及断层带内的介质，它们破坏时，微破裂在层面（节理面等软弱结构面）内有优势，以致最后的宏观破裂将沿层面发生。这类介质在强度上是强烈各向异性的，破坏的形式为沿层面的剪切破裂和垂直软弱结构面的张（拉）破裂，可采用具有软化特性的弹塑性模型来描述这种介质的性质。取层面的法线方向为 z 轴方向，层状材料的屈服条件可用层面内的应力分量 $\tau_{zx},\tau_{zy},\sigma_z$ 表示，即

$$f(\boldsymbol{\sigma},\kappa)=(\tau_{zx}^2+\tau_{zy}^2+a^2c^2)^{1/2}+\mu\sigma_z-c=0 \tag{5.95}$$

式中，a^2 是一个小参数，引入该量的目的在于用双曲线屈服面代替具有奇异顶点的库仑屈服面。c 和 μ 分别为层面内的内聚力和摩擦系数，一般为内变量 κ 的函数。如果取 $\kappa=\theta^p$，容易导出，

$$\frac{\partial f}{\partial\boldsymbol{\sigma}}=\{0,0,\mu,\tau_{zy}/\beta,\tau_{zx}/\beta,0\}^T \tag{5.96}$$

$$\boldsymbol{D}\frac{\partial f}{\partial\boldsymbol{\sigma}}=\left\{\left(K-\frac{2}{3}G\right)\mu,\left(K-\frac{2}{3}G\right)\mu,\left(K+\frac{4}{3}G\right)\mu,G\frac{\tau_{zy}}{\beta},G\frac{\tau_{zx}}{\beta},0\right\}^T \tag{5.97}$$

$$A=\left(K+\frac{4}{3}G\right)\mu^2+G\frac{\beta^2-a^2c^2}{\beta^2}+\mu(c'-\mu'\sigma_z)-a^2cc'\mu/\beta \tag{5.98}$$

式中，

$$c'=\frac{\partial c}{\partial\theta^p} \qquad \mu'=\frac{\partial u}{\partial\theta^p}$$

$$\beta=(\tau_{zx}^2+\tau_{zy}^2+a^2c^2)^{1/2}$$

利用上式容易导出塑性矩阵 $\boldsymbol{D}_p$。

5.6 粘弹性介质的本构关系

本章前述各节中考察的弹性变形和塑性变形都是与时间无关的瞬时发生的变形，在很多情况下，例如在室温条件下钢的变形，时间因素的影响是很小的，以致可以忽略不计。此时弹性理论或弹塑性理论能够合理地使用。然而在另一些情况下，时间效应的影响是很大的，甚至是根本的，例如高温环境中的金属材料，常温状态的混凝土、岩土等。在这些情况下，考虑变形过程中时间因素的影响是完全必要的。介质的变形和应力随时间变化的这种特性称

为粘性。

5.6.1　单轴应力下的粘弹性本构关系

在粘弹性体的初等理论中，通常把粘弹性体简化为由弹性元件和阻尼元件（如图 5.8 所示）组合而成的模型，再建立它的本构方程。弹性元件和阻尼元件本身的本构关系为

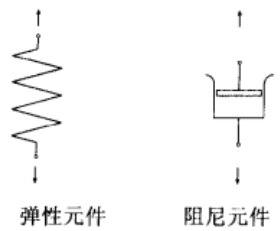


图 5.8　弹性和阻尼元件

$$\sigma = E\varepsilon \tag{5.99}$$

$$\sigma = \eta\dot{\varepsilon} \qquad \text{或} \varepsilon = \int_0^t \frac{\sigma}{\eta}\mathrm{d}\tau \tag{5.100}$$

式中，E 为弹性系数，η 为粘性系数。将弹性元件和粘性元件组合，可得到所谓的粘弹性介质。将弹性元件和粘性元件并联（图 5.9（a））就得到了著名的 Kelvin 粘弹性模型，串联（图 5.9（b））得到著名的 Maxwell 模型。但 Kelvin 模型和 Maxwell 模型都不能很好地模拟粘弹性固体材料的变形特点。目前比较适合固体材料且简单实用的粘弹性模型为三参数标准线性粘弹性模型，如图 5.10 所示。该模型包含了常用的 Maxwell 模型和 Kelvin 模型。三参数标准线性粘弹性模型满足如下关系，

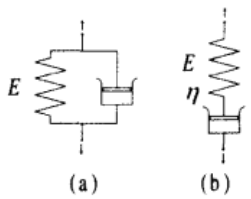


图 5.9　粘弹性模型

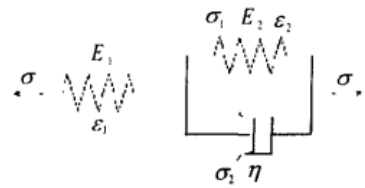


图 5.10　标准三参数模型

$$\begin{cases} \sigma = E_1\varepsilon_1 \\ \sigma_1 = E_2\varepsilon_2 \\ \sigma_2 = \eta\dot{\varepsilon}_2 \end{cases} \qquad \begin{aligned} &\sigma = \sigma_1 + \sigma_2 \\ &\varepsilon = \varepsilon_1 + \varepsilon_2 \end{aligned} \tag{5.101}$$

由（5.101）式可导出微分型本构关系，

$$\sigma + P_1\dot{\sigma} = q_0\varepsilon + q_1\dot{\varepsilon} \tag{5.102}$$

式中，

$$P_1 = \frac{\eta}{E_1 + E_2} \qquad q_0 = \frac{E_1E_2}{E_1 + E_2} \qquad q_1 = \frac{E_1}{E_1 + E_2}\eta$$

求解微分方程（5.102）式，得

$$\sigma = g_0\varepsilon(t) + \int_0^t g_1 e^{-\beta(t-\tau)}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\mathrm{d}\tau \tag{5.103}$$

式中，

$$g_0 = \frac{E_1E_2}{E_1 + E_2} \qquad g_1 = E_1 - \frac{E_1E_2}{E_1 + E_2} \qquad \beta = \frac{E_1 + E_2}{\eta} \tag{5.104}$$

称式（5.103）为积分松弛型本构关系。由图 5.10 可以看出，应力 σ 作用下的瞬时应变响应为 σ/E_1，因此称 $E_1 = g_0 + g_1$ 为短期模量。应力 σ 作用下的长期应变为 $\sigma(E_1 + E_2)/(E_1E_2)$，因此

称 $\dfrac{E_1E_2}{E_1+E_2}=g_0$ 为长期模量。

5.6.2 单轴应力下的切线模量

把式（5.103）中的时间变量进行离散，记时间分点为 $0=t_0<t_1<\cdots<t_n<t_{n+1}<\cdots$，假设 t_n 时刻的应力和应变已知，分别为 σ^n 和 ε^n。则 t_{n+1} 时刻有

$$\sigma^{n+1}=g_0\varepsilon^{n+1}+g_1h^{n+1} \tag{5.105}$$

式中，

$$h^{n+1}=\int_0^{t_{n+1}}\mathrm{e}^{-\beta(t_{n+1}-\tau)}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\mathrm{d}\tau \tag{5.106}$$

假设在时间段 $[t_n,t_{n+1}]$ 内，有

$$\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\approx\frac{\varepsilon^{n+1}-\varepsilon^n}{t_{n+1}-t_n}=\frac{\Delta\varepsilon^n}{\Delta t_n}$$

式中，$\Delta\varepsilon^n=\varepsilon^{n+1}-\varepsilon^n$ 为应变增量。则

$$\begin{aligned}h^{n+1}&=\int_0^{t_n}\mathrm{e}^{-\beta(t_{n+1}-\tau)}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\mathrm{d}\tau+\int_{t_n}^{t_{n+1}}\mathrm{e}^{-\beta(t_{n+1}-\tau)}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\mathrm{d}\tau\\&=\mathrm{e}^{-\beta\Delta t_n}\int_0^{t_n}\mathrm{e}^{-\beta(t_n-\tau)}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau}\mathrm{d}\tau+\frac{\Delta\varepsilon^n}{\Delta t_n}\int_0^{\Delta t_n}\mathrm{e}^{-\beta(\Delta t_n-\tau)}\mathrm{d}\tau\end{aligned}$$

由此容易导出 h^{n+1} 的递推关系，

$$h^{n+1}=h^n\mathrm{e}^{-\beta\Delta t_n}+\frac{\Delta\varepsilon^n}{\beta\Delta t_n}(1-\mathrm{e}^{-\beta\Delta t_n}) \tag{5.107}$$

由（5.105）式可得增量本构关系，

$$\Delta\sigma^n=g_0\Delta\varepsilon^n+g_1(h^{n+1}-h^n) \tag{5.108}$$

式中，$\Delta\sigma^n=\sigma^{n+1}-\sigma^n$ 为应力增量。为了推导出切线模量，必须将 $(h^{n+1}-h^n)$ 进一步简化。由（5.107）式可得

$$h^{n+1}-h^n=h^n(\mathrm{e}^{-\beta\Delta t_n}-1)+\frac{\Delta\varepsilon^n}{\beta\Delta t_n}(1-\mathrm{e}^{-\beta\Delta t_n})$$

当 Δt_n 充分小时，上式可简化为

$$h^{n+1}-h^n\approx Q_2\Delta\varepsilon^n \tag{5.109}$$

式中，

$$Q_2=\frac{1-\mathrm{e}^{-\beta\Delta t_n}}{\beta\Delta t_n}$$

把（5.109）式代入（5.108）式可得

$$\Delta\sigma^n=(g_0+g_1Q_2)\Delta\varepsilon^n \tag{5.110}$$

5.6.3　三维情形的本构关系

对于各向同性弹性材料，本构方程可以表示为

$$s_{ij}=2Ge_{ij}\qquad\qquad \sigma_{kk}=3K\varepsilon_{kk}$$

式中，e_{ij} 为应变张量 ε_{ij} 的偏张量。将（5.103）式推广，可得到三维各向同性粘弹性材料的本构关系，

$$s_{ij}=2G_0e_{ij}(t)+\int_0^t 2G_1\mathrm{e}^{-\alpha(t-\tau)}\frac{\mathrm{d}e_{ij}}{\mathrm{d}\tau}\mathrm{d}\tau \tag{5.111}$$

$$\sigma_{kk}=3K_0\varepsilon_{kk}(t)+\int_0^t 3K_1\mathrm{e}^{-\beta(t-\tau)}\frac{\mathrm{d}\varepsilon_{kk}}{\mathrm{d}\tau}\mathrm{d}\tau \tag{5.112}$$

上式中，G_0,G_1,K_0,K_1,α 和 β 均为材料常数，由试验确定。G_0 和 K_0 分别代表长期剪切和体积模量，G_0+G_1 和 K_0+K_1 分别代表短期剪切和体积模量，α 和 β 则分别反映了剪切和体积变形的粘性性质，值越大说明对过去的历史忘却得越快。

由上式不难得出增量形式的本构关系，

$$\Delta S_{ij}=2(G_0+G_1\frac{1-\mathrm{e}^{-\alpha\Delta t}}{\alpha\Delta t})\Delta e_{ij} \tag{5.113}$$

$$\Delta\sigma_{kk}=3(K_0+K_1\frac{1-\mathrm{e}^{-\beta\Delta t}}{\beta\Delta t})\Delta\varepsilon_{kk} \tag{5.114}$$

其他公式的推导与简单应力状态完全相同，这里不再重复。在实际工程应用中，通常假定体积变形是完全弹性的，此时有 $\beta\to\infty$。

5.7　Perzyna（波任纳）弹粘塑性材料

所谓弹粘塑性是粘弹性物质中应力达到屈服条件时开始呈现塑性变形的现象。屈服前呈现粘弹性、屈服后呈现粘性和塑性变形，特殊情况下屈服前呈现弹性变形、屈服后呈现粘性和塑性变形，这样的物质称为弹粘塑性物质。高分子聚合物、生物软组织材料和高温下的金属材料等都属于弹粘塑性材料。

5.7.1　Perzyna 弹粘塑性材料的本构方程

假设总应变可分解为弹性应变 $\boldsymbol{\varepsilon}^{e}$ 和粘塑性应变 $\boldsymbol{\varepsilon}^{vp}$ 两部分，即

$$\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}^{e}+\boldsymbol{\varepsilon}^{vp} \tag{5.115}$$

应力和弹性应变间仍满足虎克定律，

$$\boldsymbol{\sigma}=\boldsymbol{D}\boldsymbol{\varepsilon}^{e}\quad\text{或}\quad\boldsymbol{\varepsilon}^{e}=\boldsymbol{C}\boldsymbol{\sigma} \tag{5.116}$$

粘塑性变形由粘塑性屈服函数控制，等向强化的屈服条件为

$$f(\boldsymbol{\sigma},\mathrm{K})=0 \tag{5.117}$$

采用粘塑性流动法则，

$$\dot{\boldsymbol{\varepsilon}}^{vp}=\gamma<\Phi(\frac{f}{f_0})>\frac{\partial f}{\partial\boldsymbol{\sigma}} \tag{5.118}$$

式中，f_0为一与内变量κ相关的常数。

$$<\Phi(x)>=\begin{cases}\Phi(x) & 当x\geq 0\\ 0 & 当x<0\end{cases}$$

联立（5.115）、（5.116）和（5.118）式可得

$$\dot{\varepsilon}=\boldsymbol{C}\dot{\sigma}+\gamma<\Phi(\frac{f}{f_0})>\frac{\partial f}{\partial \boldsymbol{\sigma}} \tag{5.119}$$

在单轴应力下，上式可简化为

$$\dot{\varepsilon}=\frac{\dot{\sigma}}{E}+\gamma<\Phi(\frac{|\sigma|-\sigma_y}{f_0})>\frac{\partial f}{\partial \sigma}$$

在文献中，可以见到$\Phi(x)$的许多具体形式，如 Malvern 提出，

$$\Phi(x)=x \qquad \Phi(x)=e^{\lambda x}-1$$

式中，$\lambda>0$为材料常数。

Kudjanov 建议，

$$\Phi(x)=x^N \tag{5.120}$$

由于这些简单函数算出的结果往往与实验符合得不好，因而通常需要采用下列指数函数或幂函数的级数形式，

$$\Phi(x)=\sum_{k=1}^{n}a_k(e^{kx}-1) \qquad \Phi(x)=\sum_{k=1}^{n}b_k x^k$$

5.7.2 应变增量和应力增量的关系

取粘塑性的屈服函数为理想材料的 Mises 屈服函数，

$$f=\sqrt{J_2}-\mathrm{K}_0 \quad f_0=\mathrm{K}_0 \tag{5.121}$$

对于增量步$[t_n,t_{n+1}]$, $\Delta t_n=t_{n+1}-t_n$，采用隐式格式的假设，

$$\Delta(\boldsymbol{\varepsilon}^{vp})^n=\Delta t_n[(1-\alpha)(\dot{\boldsymbol{\varepsilon}}^{vp})^n+\alpha(\dot{\boldsymbol{\varepsilon}}^{vp})^{n+1}] \tag{5.122}$$

可以证明，当$\alpha\geq 1/2$时积分格式无条件稳定。为得到粘塑性应变增量和应力增量的关系，对$(\dot{\varepsilon}^{vp})^{n+1}$进行 Taylor 展开，略去高阶项可得

$$(\dot{\boldsymbol{\varepsilon}}^{vp})^{n+1}=(\dot{\boldsymbol{\varepsilon}}^{vp})^n+\left(\frac{\partial \dot{\boldsymbol{\varepsilon}}^{vp}}{\partial \boldsymbol{\sigma}}\right)_n\Delta\boldsymbol{\sigma}^n \tag{5.123}$$

简记为

$$(\dot{\boldsymbol{\varepsilon}}^{vp})^{n+1}=(\dot{\boldsymbol{\varepsilon}}^{vp})^n+\boldsymbol{H}^n\Delta\boldsymbol{\sigma}^n \tag{5.124}$$

把上式代入（5.122）式，可得

$$\Delta(\boldsymbol{\varepsilon}^{vp})^n=(\dot{\boldsymbol{\varepsilon}}^{vp})^n\Delta t_n+\alpha\boldsymbol{H}^n\Delta t_n\Delta\boldsymbol{\sigma}^n \tag{5.125}$$

简记为

$$\Delta(\boldsymbol{\varepsilon}^{vp})^n=(\dot{\boldsymbol{\varepsilon}}^{vp})^n\Delta t_n+\boldsymbol{C}^n\Delta\boldsymbol{\sigma}_n \tag{5.126}$$

由（5.115）式和（5.116）式可得

$$\Delta\boldsymbol{\sigma}^n = \boldsymbol{D}\Delta(\boldsymbol{\varepsilon}^e)^n = \boldsymbol{D}(\Delta\boldsymbol{\varepsilon}^n - \Delta(\boldsymbol{\varepsilon}^{vp})^n)$$

把（5.126）式代入上式可得

$$\Delta\boldsymbol{\sigma}^n = \boldsymbol{D}(\Delta\boldsymbol{\varepsilon}^n - (\dot{\boldsymbol{\varepsilon}}^{vp})^n \Delta t_n - \boldsymbol{C}^n \Delta\boldsymbol{\sigma}^n)$$

解之得

$$\Delta\boldsymbol{\sigma}^n = (\boldsymbol{D}\boldsymbol{C}^n + \boldsymbol{I})^{-1}\boldsymbol{D}(\Delta\boldsymbol{\varepsilon}^n - (\dot{\boldsymbol{\varepsilon}}^{vp})^n \Delta t_n)$$

因为

$$(\boldsymbol{D}\boldsymbol{C}^n + \boldsymbol{I})^{-1}\boldsymbol{D} = (\boldsymbol{D}\boldsymbol{C}^n + \boldsymbol{I})^{-1}\boldsymbol{C}^{-1} = [\boldsymbol{C}(\boldsymbol{D}\boldsymbol{C}^n + \boldsymbol{I})]^{-1} = [\boldsymbol{C}^n + \boldsymbol{C}]^{-1} = \boldsymbol{D}^n$$

所以

$$\Delta\boldsymbol{\sigma}^n = \boldsymbol{D}^n(\Delta\boldsymbol{\varepsilon}^n - (\dot{\boldsymbol{\varepsilon}}^{vp})^n \Delta t_n) \tag{5.127}$$

式中，

$$\boldsymbol{D}^n = [\boldsymbol{C}^n + \boldsymbol{C}]^{-1}$$

把（5.121）式代入（5.118）式，得

$$\dot{\boldsymbol{\varepsilon}}^{vp} = \gamma < \Phi(\frac{f}{f_0}) > \frac{1}{2\sqrt{J_2}}\bar{\boldsymbol{s}} \tag{5.128}$$

$$\boldsymbol{H} = \frac{\partial\dot{\boldsymbol{\varepsilon}}^{vp}}{\partial\boldsymbol{\sigma}} = \gamma\left[< \Phi(\frac{f}{f_0}) > \frac{\partial(\frac{\bar{\boldsymbol{s}}}{2\sqrt{J_2}})}{\partial\boldsymbol{\sigma}} + \frac{\mathrm{d} < \Phi(\frac{f}{f_0}) > \partial(\frac{f}{f_0})}{\mathrm{d}(\frac{f}{f_0})\quad\partial\boldsymbol{\sigma}} \cdot \frac{\bar{\boldsymbol{s}}^T}{2\sqrt{J_2}}\right] \tag{5.129}$$

其中，

$$\frac{\partial(\frac{\bar{\boldsymbol{s}}}{2\sqrt{J_2}})}{\partial\boldsymbol{\sigma}} = \frac{1}{2\sqrt{J_2}}\frac{\partial\bar{\boldsymbol{s}}}{\partial\boldsymbol{\sigma}} - \frac{1}{4J_2^{\frac{3}{2}}}\frac{\partial J_2}{\partial\boldsymbol{\sigma}}\bar{\boldsymbol{s}}^T = \frac{1}{2\sqrt{J_2}}\left[\frac{\partial\bar{\boldsymbol{s}}}{\partial\boldsymbol{\sigma}} - \frac{1}{2J_2}\bar{\boldsymbol{s}}\,\bar{\boldsymbol{s}}^T\right] \tag{5.130}$$

为方便起见，记

$$\boldsymbol{M}_1 = \frac{\partial\bar{\boldsymbol{s}}}{\partial\boldsymbol{\sigma}} \qquad \boldsymbol{M}_2 = \bar{\boldsymbol{s}}\,\bar{\boldsymbol{s}}^T$$

容易直接算出，

$$\boldsymbol{M}_1 = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & & & \\ & \frac{2}{3} & -\frac{1}{3} & & 0 & \\ & & \frac{2}{3} & & & \\ & \text{对称} & & 2 & & \\ & & & & 2 & \\ & & & & & 2 \end{bmatrix}$$

把（5.130）式代入（5.129）式，可得

$$H=\gamma\left[\Phi<(\frac{f}{f_0})>\frac{1}{2\sqrt{J_2}}(\boldsymbol{M}_1-\frac{1}{2J_2}\boldsymbol{M}_2)+\frac{1}{f_0}\frac{\mathrm{d}<\Phi(\frac{f}{f_0})>}{\mathrm{d}(\frac{f}{f_0})}(\frac{1}{2\sqrt{J_2}})^2\boldsymbol{M}_2\right]$$

$$=\gamma\frac{<\Phi(\frac{f}{f_0})>}{2\sqrt{J_2}}\boldsymbol{M}_1+\gamma\left[\frac{1}{4f_0J_2}\frac{\mathrm{d}<\Phi(\frac{f}{f_0})>}{\mathrm{d}(\frac{f}{f_0})}-\frac{1}{4J_2^{3/2}}<\Phi(\frac{f}{f_0})>\right]\boldsymbol{M}_2$$

$$=P_1\boldsymbol{M}_1+P_2\boldsymbol{M}_2$$

式中，

$$P_1=\gamma\frac{<\Phi(\frac{f}{f_0})>}{2\sqrt{J_2}}\qquad P_2=\gamma\left[\frac{1}{4f_0J_2}\frac{\mathrm{d}<\Phi(\frac{f}{f_0})>}{\mathrm{d}(\frac{f}{f_0})}-\frac{1}{4J_2^{3/2}}<\Phi(\frac{f}{f_0})>\right]$$

参考文献

[1] 匡震邦.非线性连续介质力学基础. 西安:西安交通大学出版社，1989

[2] 黄克智，黄永刚编著.固体本构关系. 北京:清华大学出版社，1999

[3] 丁皓江等著.横观各向同性弹性力学.杭州:浙江大学出版社， 1997

[4] 徐兴，郭乙木，沈永兴.非线性有限元及程序设计. 杭州:浙江大学出版社，1993

第6章

线性有限单元法

前面5章系统地介绍了固体力学问题的基本方程，包括应力描述、变形描述和运动描述。从本章开始，我们将系统地介绍固体力学问题中的有限单元法。由于非线性问题的有限单元分析总是归结为一系列线性问题的分析。因此可以说，线性有限单元法是非线性有限单元法的基础。这一章对线性问题的有限单元分析作一概述，详细的相关内容可参见[1]~[4]。

6.1 线弹性力学问题的基本方程

在小变形条件下，现时构形和初始构形的区别很小，可以忽略不计。从前面的章节很容易得到线弹性力学问题的基本方程。由于有限元法经常用到矩阵表达形式，现将该形式综合于下，但不作详细推导。

一、运动方程

弹性体V域内任意一点沿三个坐标轴x、y、z方向的平衡方程为

$$\begin{aligned}
&\frac{\partial \sigma_x}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}+f_x=\rho\ddot{u}\\
&\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \sigma_y}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}+f_y=\rho\ddot{v}\\
&\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{yz}}{\partial y}+\frac{\partial \sigma_z}{\partial z}+f_z=\rho\ddot{w}
\end{aligned} \tag{6.1}$$

式中，f_x、f_y和f_z分别为沿x、y、z方向单位体积的体积力，u、v和w分别为沿x、y、z方向的位移，ρ为质量密度。(6.1)可以写成如下矩阵形式

$$\boldsymbol{L}\cdot\boldsymbol{\sigma}+\boldsymbol{f}=\rho\ddot{\boldsymbol{u}} \tag{6.2}$$

式中，$\boldsymbol{L}$为微分算子，

$$\boldsymbol{L}=\begin{bmatrix}
\frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y}\\
0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x}\\
0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0
\end{bmatrix}$$

应力矢量$\boldsymbol{\sigma}$、体积力矢量$\boldsymbol{f}$和位移矢量$\boldsymbol{u}$分别为

$$\boldsymbol{\sigma}=\left\{\sigma_x,\sigma_y,\sigma_z,\tau_{yz},\tau_{yx},\tau_{xy}\right\}^T$$

$$\boldsymbol{f}=\left\{f_x,f_y,f_z\right\}^T$$

$$\boldsymbol{u}=\left\{u,v,w\right\}^T$$

二、几何方程——应变-位移关系

在微小变形的情况下，略去位移导数的高次幂，可得应变和位移矢量的关系式，

$$\begin{aligned}&\varepsilon_x=\partial u/\partial x \qquad \varepsilon_y=\partial v/\partial y \qquad \varepsilon_z=\partial w/\partial z\\&\gamma_{yz}=\partial v/\partial z+\partial w/\partial y \quad \gamma_{zx}=\partial u/\partial z+\partial w/\partial x \quad \gamma_{xy}=\partial u/\partial y+\partial v/\partial x\end{aligned}\tag{6.3}$$

注意工程应变分量和应变张量分量的不同定义，上式的矩阵形式为

$$\boldsymbol{\varepsilon}=\boldsymbol{L}^T\cdot\boldsymbol{u}\tag{6.4}$$

式中应变矢量

$$\boldsymbol{\varepsilon}=\left\{\varepsilon_x,\varepsilon_y,\varepsilon_z,\gamma_{yz},\gamma_{zx},\gamma_{xy}\right\}^T$$

三、物理方程——应力应变关系

弹性力学中应力应变间的转换关系也称弹性关系，用矩阵形式可表示为

$$\boldsymbol{\sigma}=\boldsymbol{D}\cdot\boldsymbol{\varepsilon}\tag{6.5}$$

式中，$\boldsymbol{D}$ 称为弹性矩阵。对于各向同性弹性材料，它完全取决于弹性模量 E 和泊桑比 v。

$$\boldsymbol{D}=\frac{E(1-v)}{(1+v)(1-2v)}\begin{bmatrix}1 & \frac{v}{1-v} & \frac{v}{1-v} & & & \\ & 1 & \frac{v}{1-v} & & 0 & \\ & & 1 & & & \\ & \text{对称} & & \frac{1-2v}{2(1-v)} & & \\ & & & & \frac{1-2v}{2(1-v)} & \\ & & & & & \frac{1-2v}{2(1-v)}\end{bmatrix}\tag{6.6}$$

也可以采用拉梅(Lame)常数 λ 和 μ 来表征弹性体的弹性常数，

$$\lambda=\frac{Ev}{(1+v)(1-2v)} \qquad \mu=\frac{E}{2(1+v)}\tag{6.7}$$

其中，常数 μ 也称为剪切弹性模量，通常用符号 G 来表示。物理方程中的弹性矩阵 $\boldsymbol{D}$ 亦可表示为

$$\boldsymbol{D}=\begin{bmatrix}\lambda+G & \lambda & \lambda & & & \\ & \lambda+G & \lambda & & & \\ & & \lambda+G & & & \\ & & & G & & \\ & \text{对称} & & & G & \\ & & & & & G\end{bmatrix}\tag{6.8}$$

对（6.5）式求逆可得物理方程的另一种表达形式，

$$\boldsymbol{\varepsilon}=\boldsymbol{C}\boldsymbol{\sigma} \tag{6.9}$$

式中，$\boldsymbol{C}$ 是柔度矩阵。

四、边界条件

记弹性体 V 的全部边界为 Γ。一部分边界上已知外力 $\bar{t}_x,\bar{t}_y,\bar{t}_z$，称为力的边界条件，这部分边界用 Γ_t 表示；另一部分边界上弹性体的位移 $\bar{u}$、$\bar{v}$ 和 $\bar{w}$ 已知，称为几何边界条件或位移边界条件，这部分边界用 Γ_u 表示。这两部分边界互不重叠，共同构成弹性体的全部边界，即

$$\Gamma_u\cap\Gamma_t=\phi\quad\Gamma_u\cup\Gamma_t=\Gamma \tag{6.10}$$

设给定力的边界外法线方向的单位方向矢量为 $\{n_x,n_y,n_z\}^T$，则由四面体的平衡条件可得力的边界条件为

$$\begin{cases}\bar{t}_x=\sigma_x n_x+\tau_{xy}n_y+\tau_{xz}n_z\\ \bar{t}_y=\tau_{xy}n_x+\sigma_y n_y+\tau_{yz}n_z\\ \bar{t}_z=\tau_{xz}n_x+\tau_{yz}n_y+\sigma_z n_z\end{cases} \tag{6.11}$$

写成矩阵形式

$$\bar{\boldsymbol{t}}=\boldsymbol{N}_t\,\boldsymbol{\sigma} \tag{6.12}$$

式中，

$$\boldsymbol{N}_t=\begin{bmatrix}n_x&0&0&0&n_z&n_y\\0&n_y&0&n_z&0&n_x\\0&0&n_z&n_y&n_x&0\end{bmatrix} \tag{6.13}$$

在给定位移的边界上，

$$u=\bar{u},\ v=\bar{v},\ w=\bar{w} \tag{6.14}$$

写成矩阵形式为

$$\boldsymbol{u}=\bar{\boldsymbol{u}} \tag{6.15}$$

对于平面问题和轴对称问题，也可类似地得到控制方程和边界条件，简单归纳如下：

平面问题

位移矢量　$\boldsymbol{u}=\{u,v\}^T$　(6.16)

应变矢量　$\boldsymbol{\varepsilon}=\{\varepsilon_x,\varepsilon_y,\gamma_{xy}\}^T=\boldsymbol{L}^T\boldsymbol{u}$　(6.17)

应力矢量　$\boldsymbol{\sigma}=\{\sigma_x,\sigma_y,\tau_{xy}\}^T=\boldsymbol{D}\boldsymbol{\varepsilon}$　(6.18)

微分算子　$$\boldsymbol{L}=\begin{bmatrix}\dfrac{\partial}{\partial x}&0&\dfrac{\partial}{\partial y}\\0&\dfrac{\partial}{\partial y}&\dfrac{\partial}{\partial x}\end{bmatrix}$$　(6.19)

运动方程　$\boldsymbol{L}\boldsymbol{\sigma}+\boldsymbol{f}=\rho\ddot{\boldsymbol{u}}$　(6.20)

体积力矢量　$\boldsymbol{f}=\{f_x,f_y\}^T$　(6.21)

弹性矩阵
$$\boldsymbol{D}=\frac{E_0}{1-\nu_0^2}\begin{bmatrix}1 & \nu_0 & 0\\ \text{对称} & 1 & 0\\ & & \dfrac{1-\nu_0}{2}\end{bmatrix} \tag{6.22}$$

平面应力问题 $E_0=E \quad \nu_0=\nu$

平面应变问题 $E_0=E/(1-\nu^2) \quad \nu_0=\nu/(1-\nu)$

力边界条件 $\bar{\boldsymbol{t}}=\boldsymbol{N}_t\boldsymbol{\sigma} \qquad \boldsymbol{N}_t=\begin{bmatrix}n_x & 0 & n_y\\ 0 & n_y & n_x\end{bmatrix} \qquad \bar{\boldsymbol{t}}=\begin{Bmatrix}\bar{t}_x\\ \bar{t}_y\end{Bmatrix}$

位移边界条件 $u=\bar{u} \qquad v=\bar{v}$

轴对称问题

位移矢量 $$\boldsymbol{u}=\{u,w\}^T \tag{6.23}$$

应变矢量 $$\boldsymbol{\varepsilon}=\{\varepsilon_r,\varepsilon_z,\gamma_{rz},\varepsilon_\theta\}^T=\boldsymbol{L}^T\boldsymbol{u} \tag{6.24}$$

应力矢量 $$\boldsymbol{\sigma}=\{\sigma_r,\sigma_z,\tau_{rz},\sigma_\theta\}^T=\boldsymbol{D}\boldsymbol{\varepsilon} \tag{6.25}$$

微分算子
$$\boldsymbol{L}=\begin{bmatrix}\dfrac{\partial}{\partial r} & 0 & \dfrac{\partial}{\partial z} & \dfrac{1}{r}\\ 0 & \dfrac{\partial}{\partial z} & \dfrac{\partial}{\partial r} & 0\end{bmatrix} \tag{6.26}$$

弹性矩阵
$$\boldsymbol{D}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix}1 & \dfrac{\nu}{1-\nu} & 0 & \dfrac{\nu}{1-\nu}\\ & 1 & 0 & \dfrac{\nu}{1-\nu}\\ & \text{对称} & \dfrac{1-2\nu}{2(1-\nu)} & 0\\ & & & 1\end{bmatrix} \tag{6.27}$$

力边界条件 $\bar{\boldsymbol{t}}=\boldsymbol{N}_t\boldsymbol{\sigma} \qquad \boldsymbol{N}_t=\begin{bmatrix}n_x & 0 & n_z\\ 0 & n_z & n_x\end{bmatrix} \qquad \bar{\boldsymbol{t}}=\begin{Bmatrix}\bar{t}_x\\ \bar{t}_z\end{Bmatrix}$

位移边界条件 $u=\bar{u} \qquad w=\bar{w}$

6.2 线弹性有限单元法概述

用有限单元法求解满足边界条件的位移和应力分布基本上分三大步进行：结构的离散化、单元分析和总体分析。

6.2.1 结构的离散化

结构的离散化是有限元法分析的第一步，是有限元分析的基础。所谓结构的离散化就是用有限个按一定规则在节点和边界上满足一定关系的离散单元的组合体代替原来的结构，这

样的组合体称为有限单元系统。如果待分析的对象是杆系结构，那么单元的剖分是简单而显而易见的。如果分析的对象是连续体，为有效逼近实际结构，必须认真考虑单元的形状、剖分方法、单元尺寸等问题。以平面问题为例，假设用最简单、最常用的三角形单元，把弹性体分割成有限个互不重叠的三角形，并指定三角形顶点为节点，构成一个单元集合体，代替原来的弹性体。此时，弹性体的边界为一系列折线所代替，随着单元的细分，单元集合体不断地逼近弹性体。

设有限单元系统共有 NE 个单元，NP 个节点。在有限元的位移法中，系统的节点位移是基本未知量。记第 k 个节点的位移矢量为 $\boldsymbol{u}_k$，对于一般的三维问题，

$$\boldsymbol{u}_k = \{u_k, v_k, w_k\}^T$$

对于平面问题和轴对称问题，

$$\boldsymbol{u}_k = \{u_k, v_k\}^T$$

则有限元系统的节点位移矢量可以表示为

$$\boldsymbol{d} = \{\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \cdots, \boldsymbol{u}_{NP}^T\}^T \tag{6.28}$$

连续体离散化为单元体集合，其本质是剖分插值，即将弹性体所占的域剖分成称为单元的子域的集合，将未知位移场用分片插值函数近似。

6.2.2　单元分析

单元分析是有限元分析中最重要的步骤之一，它通常包括单元位移模式的选择和单元力学特性分析等内容。

一、位移插值、插值形函数

为了能够用节点位移表示单元位移、应变和应力，必须对单元位移的分布作出一定的假设，也就是假设位移是坐标的某种简单函数(通常为关于坐标的代数多项式)，这种函数就称为位移模式。选择恰当的位移模式是有限单元法分析中的关键。一旦确定了单元模式，单元内任意点的位移就可以近似地由该单元各节点位移来表示，

$$\boldsymbol{u} = \boldsymbol{N} \cdot \boldsymbol{d}^e \tag{6.29}$$

式中，$\boldsymbol{u}$ 为单元内任意点的位移矢量，$\boldsymbol{d}^e$ 为单元所有节点的位移矢量，$\boldsymbol{N}$ 为位移插值形函数矩阵。

合理的位移插值形函数的选取十分重要，它直接关系到有限元法的收敛性。一般要求 $\boldsymbol{u}$ 在单元内和单元间都是连续的。龙驭球等弱化了单元间位移协调的要求，提出了广义协调单元的概念，并构造了多种广义协调单元。本章仅限于讨论协调单元，以下给出几种常用单元的插值形函数。

(1) 三角形平面单元

以二维线性函数作为位移模式，位移插值形函数矩阵为

$$\boldsymbol{N} = [\boldsymbol{N}_1, \boldsymbol{N}_2, \boldsymbol{N}_3] \qquad \boldsymbol{N}_i = N_i \boldsymbol{I}_{2\times 2} \tag{6.30}$$

式中，$\boldsymbol{I}_{2\times 2}$ 为 2×2 的单位矩阵。

$$N_1=\frac{1}{2\Delta}\begin{vmatrix}1 & x & y\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{vmatrix}\qquad N_2=\frac{1}{2\Delta}\begin{vmatrix}1 & x_1 & y_1\\ 1 & x & y\\ 1 & x_3 & y_3\end{vmatrix}\qquad N_3=\frac{1}{2\Delta}\begin{vmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x & y\end{vmatrix}$$

其中，

$$2\Delta=\begin{vmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{vmatrix}$$

从解析几何知，式中 Δ 等于三角形单元的面积，为使该面积的计算值不致为负，节点 1、2、3 的次序必须是逆时针转向，如图 6.1 所示。在任意点 (x, y) 处，N_i 的值等于该点与节点 i 相对的边构成的三角形面积和单元面积的比值。因此，人们习惯将 N_1、N_2 和 N_3 写为 L_1、L_2 和 L_3，并称之为面积坐标。显然，形函数 N_i 在节点 i 处的值等于 1，在其他节点处的值为零。实际上，N_i 在节点 i 所对边上处处为零。在任意点，三个形函数的和总是等于 1 的。

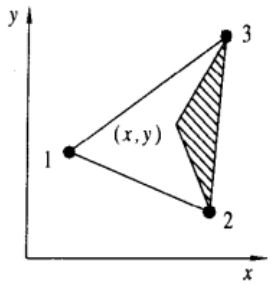


图 6.1 三角形平面单元

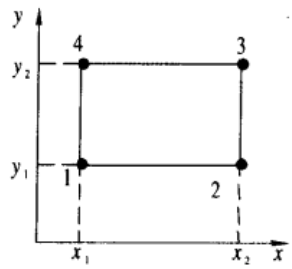


图 6.2 矩形平面单元

(2) 矩形平面单元

如图 6.2 所示矩形平面单元，以双线性函数作为位移模式，单元位移插值形函数矩阵为

$$\boldsymbol{N}=[\boldsymbol{N}_1,\boldsymbol{N}_2,\boldsymbol{N}_3,\boldsymbol{N}_4]\qquad \boldsymbol{N}_i=N_i\boldsymbol{I}_{2\times 2}\tag{6.31}$$

其中，

$$N_i=(1+\xi_i\xi)(1+\eta_i\eta)/4$$

$$\xi=\frac{2(x-x_1)}{x_2-x_1}-1\qquad \eta=\frac{2(y-y_1)}{y_2-y_1}-1$$

$$\xi_i=\xi\big|_{x=x_i}\qquad \eta_i=\eta\big|_{y=y_i}$$

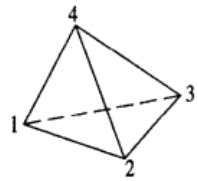


图6.3 四面体单元

(3) 四面体空间单元

如图 6.3 所示 4 节点四面体单元，以空间的线性函数为位移模式，单元位移插值形函数矩阵为

$$\boldsymbol{N}=[\boldsymbol{N}_1,\boldsymbol{N}_2,\boldsymbol{N}_3,\boldsymbol{N}_4]\qquad \boldsymbol{N}_i=N_i\boldsymbol{I}_{3\times 3}\tag{6.32}$$

式中，

$$N_1=\frac{1}{6V}\begin{vmatrix}1 & x & y & z\\ 1 & x_2 & y_2 & z_2\\ 1 & x_3 & y_3 & z_3\\ 1 & x_4 & y_4 & z_4\end{vmatrix}\qquad N_2=\frac{1}{6V}\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x & y & z\\ 1 & x_3 & y_3 & z_3\\ 1 & x_4 & y_4 & z_4\end{vmatrix}$$

$$N_3=\frac{1}{6V}\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x_2 & y_2 & z_2\\ 1 & x & y & z\\ 1 & x_4 & y_4 & z_4\end{vmatrix}\qquad N_4=\frac{1}{6V}\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x_2 & y_2 & z_2\\ 1 & x_3 & y_3 & z_3\\ 1 & x & y & z\end{vmatrix}$$

其中，

$$6V=\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x_2 & y_2 & z_2\\ 1 & x_3 & y_3 & z_3\\ 1 & x_4 & y_4 & z_4\end{vmatrix}$$

根据解析几何知识同样可以看出，V 等于四面体的体积。为保证该体积的值为正，节点 1、2、3、4 应满足右手法则。N_i 等于单元中任意点 (x,y,z) 与节点 i 所对的面构成的四面体体积跟单元体积的比值。

由式（6.30）、（6.31）、（6.32）容易看出，形函数具有如下性质：

（1）N_i 在第 i 个节点 (x_i,y_i) 或 (x_i,y_i,z_i) 处的值为 1，在其他节点处的值为 0；

（2）单元某条边（面）上的位移由该边（面）上相应节点惟一确定，与其他节点无关；

（3）单元内任意一点形函数的和等于 1。

利用上述性质，很容易证明相邻单元的位移分别进行插值以后，在公共边（面）上满足位移连续条件。

二、单元应变和应力

把式（6.29）代入几何方程（6.4）式，可求得单元中任一点的应变，

$$\boldsymbol{\varepsilon}=\boldsymbol{L}^T(\boldsymbol{N}\cdot\boldsymbol{d}^e)=\boldsymbol{B}\boldsymbol{d}^e \tag{6.33}$$

注意到，$\boldsymbol{L}$ 是坐标的线性算子，$\boldsymbol{d}^e$ 与坐标无关。对于 3 节点三角形平面单元和 4 节点四面体空间单元，$\boldsymbol{B}$ 为常数矩阵，也就是说单元应变为常应变。很明显，应变在单元内是连续的，而在单元间一般是不连续的。将单元应变代入物理方程即可获得单元应力。注意，单元间的应力也不一定连续。

三、单元积分形式的平衡方程

在小变形条件下，单元虚功方程为

$$\int_{V^e}(\delta\boldsymbol{\varepsilon})^T\boldsymbol{\sigma}\mathrm{d}V=\int_{V^e}(\delta\boldsymbol{u})^T\boldsymbol{f}\,\mathrm{d}V+\int_{\Gamma_t^e}(\delta\boldsymbol{u})^T\bar{\boldsymbol{t}}\mathrm{d}\Gamma+\int_{\Gamma^e-\Gamma_t^e-\Gamma_u^e}(\delta\boldsymbol{u})^T\hat{\boldsymbol{t}}\,\mathrm{d}\Gamma \tag{6.34}$$

式中，V^e 为单元所占据的空间区域，Γ^e 为单元表面，$\Gamma_t^e=\Gamma_t\cap\Gamma^e$，$\Gamma_u^e=\Gamma_u\cap\Gamma^e$，$\Gamma^e-\Gamma_t^e-\Gamma_t^u$ 为相邻单元间的交界面，$\hat{\boldsymbol{t}}$ 为交界面上单元间的相互作用力。

分别对(6.29)式和(6.33)式变分，并注意到 $\boldsymbol{N}$ 和 $\boldsymbol{B}$ 是空间坐标的函数，其变分为 0，可得

$$\delta\boldsymbol{\varepsilon}=\boldsymbol{B}\delta\boldsymbol{d}^e \tag{6.35}$$

$$\delta\boldsymbol{u}=\boldsymbol{N}\,\delta\boldsymbol{d}^e \tag{6.36}$$

把（6.35）式和（6.36）式代入（6.34）式，可得

$$\int_{V^e}\left(\delta \boldsymbol{d}^e\right)^T \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B}\boldsymbol{d}^e \mathrm{d}V=\int_{V^e}\left(\delta \boldsymbol{d}^e\right)^T \boldsymbol{N}^T \boldsymbol{f}\ \mathrm{d}V+\int_{\Gamma_t^e}\left(\delta \boldsymbol{d}^e\right)^T \boldsymbol{N}^T \overline{\boldsymbol{t}} \mathrm{d}\Gamma+\int_{\Gamma^e-\Gamma_t^e-\Gamma_u^e}\left(\delta \boldsymbol{d}^e\right)^T \boldsymbol{N}^T \hat{\boldsymbol{t}} \mathrm{d}\Gamma$$

注意到，$\boldsymbol{d}^e$ 与坐标无关，因此可以从积分号中提取出来，注意到 $\delta\boldsymbol{d}^e$ 是任意的，则上式可进一步变化为

$$\int_{V^e} \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B}\mathrm{d}V \boldsymbol{d}^e=\int_{V^e} \boldsymbol{N}^T \boldsymbol{f}\ \mathrm{d}V+\int_{\Gamma_t^e} \boldsymbol{N}^T \overline{\boldsymbol{t}} \mathrm{d}\Gamma+\int_{\Gamma^e-\Gamma_t^e-\Gamma_u^e} \boldsymbol{N}^T \hat{\boldsymbol{t}}\ \mathrm{d}\Gamma$$

记

$$\boldsymbol{K}^e=\int_{V^e} \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B}\mathrm{d}V \qquad \boldsymbol{F}_f^e=\int_{V^e} \boldsymbol{N}^T \boldsymbol{f}\ \mathrm{d}V$$

$$\boldsymbol{F}_t^e=\int_{\Gamma_t^e} \boldsymbol{N}^T \overline{\boldsymbol{t}} \mathrm{d}\Gamma \qquad \hat{\boldsymbol{F}}_t^e=\int_{\Gamma^e-\Gamma_t^e-\Gamma_u^e} \boldsymbol{N}^T \hat{\boldsymbol{t}}\ \mathrm{d}\Gamma$$

单元的平衡方程可以表示为

$$\boldsymbol{K}^e\boldsymbol{d}^e=\boldsymbol{P}^e=\boldsymbol{F}_f^e+\boldsymbol{F}_t^e+\hat{\boldsymbol{F}}_t^e \tag{6.37}$$

称 $\boldsymbol{K}^e$ 为单元刚度矩阵，它显然是对称的、半正定的，$\boldsymbol{F}_f^e$ 为单元的体积力等效节点力，$\boldsymbol{F}_t^e$ 为单元表面外力等效节点力，$\hat{\boldsymbol{F}}_t^e$ 为相邻单元对该单元的作用力产生的等效节点力。

集中力可以视为一种特殊的体积力或表面力。例如，单元 e 内作用在 $\boldsymbol{x}_0$ 点的集中力 $\boldsymbol{R}_0$ 可以表示为如下形式的体积力，

$$\boldsymbol{f}=\boldsymbol{R}_0\delta(\boldsymbol{x}-\boldsymbol{x}_0)$$

其中，$\delta(\boldsymbol{x})$ 为脉冲函数。集中力对应的等效节点力为

$$\boldsymbol{F}_0=\int_{V^e} \boldsymbol{N}^T \boldsymbol{R}_0\delta(\boldsymbol{x}-\boldsymbol{x}_0)\mathrm{d}V=\boldsymbol{N}(\boldsymbol{x}_0)^T \boldsymbol{R}_0 \tag{6.38}$$

特殊地，若集中力 $\boldsymbol{R}_0$ 作用在单元的第 i 个节点上，则有

$$\left(\boldsymbol{F}_0\right)_j=\boldsymbol{R}_0\delta_{ij}\quad(j=1,2,\cdots,n) \tag{6.39}$$

式中，n 为单元节点数。由（6.38）式和（6.39）式可以看出，若 $\boldsymbol{R}_0$ 作用在节点上，则 $\boldsymbol{R}_0$ 即为作用在该节点上的等效节点力；若 $\boldsymbol{R}_0$ 作用在单元某个节面内或某条节线上，则 $\boldsymbol{R}_0$ 的等效节点力作用在该面或该线上的节点，而与其他节点无关。为方便起见，通常要求集中力作用在节点上。

6.2.3 总体分析

注意到式（6.37）仅反映一个单元的平衡，且其中单元间相互作用的等效节点力 $\hat{\boldsymbol{F}}_t^e$ 是未知的。同时我们还注意到相邻单元 $\boldsymbol{e}_i$ 和 $\boldsymbol{e}_j$ 在界面上的作用是相互的，$\boldsymbol{e}_i$ 单元对 $\boldsymbol{e}_j$ 单元的作用力与 $\boldsymbol{e}_j$ 单元对 $\boldsymbol{e}_i$ 单元的作用力具有大小相等、方向相反的特点。如果单元位移模式满足单元间协调条件，两个单元合并在一起时，相互作用的等效节点力恰好抵消。为了由单元的特性矩阵得到整体有限元系统的特性，需对（6.37）式作适当的扩大改写，使得具有统一格式。由于单元 $\boldsymbol{e}$ 的节点位移矢量 $\boldsymbol{d}^e$ 是系统节点位移矢量 $\boldsymbol{d}$ 的子集，因此存在一个选择矩阵 $\boldsymbol{T}^e$，使得

$$\boldsymbol{d}^e=\boldsymbol{T}^e\boldsymbol{d} \tag{6.40}$$

将上式代入单元平衡方程式（6.37）可得

$$\boldsymbol{K}^e\boldsymbol{T}^e\boldsymbol{d}=\boldsymbol{F}_f^e+\boldsymbol{F}_t^e+\hat{\boldsymbol{F}}_t^e$$

上式两边同时乘以 $\boldsymbol{T}^e$ 的转置可得

$$\left(\boldsymbol{T}^e\right)^T\boldsymbol{K}^e\boldsymbol{T}^e\boldsymbol{d}=\left(\boldsymbol{T}^e\right)^T\left(\boldsymbol{F}_f^e+\boldsymbol{F}_t^e+\hat{\boldsymbol{F}}_t^e\right)\tag{6.41}$$

为说明（6.41）式，以图 6.4 所示的有限单元系统为例。

考虑单元③，显然有

$$\boldsymbol{d}^e=\left\{\boldsymbol{u}_{1'}^T,\boldsymbol{u}_{2'}^T,\boldsymbol{u}_{3'}^T\right\}^T$$

$$\boldsymbol{d}=\left\{\boldsymbol{u}_1^T,\boldsymbol{u}_2^T,\boldsymbol{u}_3^T,\boldsymbol{u}_4^T,\boldsymbol{u}_5^T\right\}^T$$

式中，$\boldsymbol{u}_i=\left\{u_i,v_i\right\}^T$ 为第 i 个节点的位移矢量，$\boldsymbol{u}_{i'}$ 为单元第 i 个节点（注意不一定是编号为 i 的节点）的位移矢量。显然

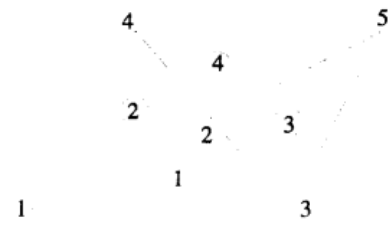


图 6.4　有限元系统示意图

$$\boldsymbol{u}_{1'}=\boldsymbol{u}_2,\quad \boldsymbol{u}_{2'}=\boldsymbol{u}_3,\quad \boldsymbol{u}_{3'}=\boldsymbol{u}_5$$

于是，

$$\boldsymbol{T}^e=\begin{bmatrix}0 & \boldsymbol{I}_{2\times2} & 0 & 0 & 0\\ 0 & 0 & \boldsymbol{I}_{2\times2} & 0 & 0\\ 0 & 0 & 0 & 0 & \boldsymbol{I}_{2\times2}\end{bmatrix}$$

为方便起见，将单元刚度阵和等效节点力矢量分块写成，

$$\boldsymbol{K}^e=\begin{bmatrix}\boldsymbol{K}_{1'1'}^e & \boldsymbol{K}_{1'2'}^e & \boldsymbol{K}_{1'3'}^e\\ \boldsymbol{K}_{2'1'}^e & \boldsymbol{K}_{2'2'}^e & \boldsymbol{K}_{2'3'}^e\\ \boldsymbol{K}_{3'1'}^e & \boldsymbol{K}_{3'2'}^e & \boldsymbol{K}_{3'3'}^e\end{bmatrix}\quad \boldsymbol{F}^e=\begin{Bmatrix}\boldsymbol{F}_{1'}^e\\ \boldsymbol{F}_{2'}^e\\ \boldsymbol{F}_{3'}^e\end{Bmatrix}$$

$$\left(\boldsymbol{T}^e\right)^T\boldsymbol{K}^e\boldsymbol{T}^e=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{1'1'}^e & \boldsymbol{K}_{1'2'}^e & \boldsymbol{0} & \boldsymbol{K}_{1'3'}^e\\ \boldsymbol{0} & \boldsymbol{K}_{2'1'}^e & \boldsymbol{K}_{2'2'}^e & \boldsymbol{0} & \boldsymbol{K}_{2'3'}^e\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{3'1'}^e & \boldsymbol{K}_{3'2'}^e & \boldsymbol{0} & \boldsymbol{K}_{3'3'}^e\end{bmatrix}\quad \left(\boldsymbol{T}^e\right)^T\boldsymbol{F}^e=\begin{Bmatrix}\boldsymbol{0}\\ \boldsymbol{F}_{1'}^e\\ \boldsymbol{F}_{2'}^e\\ \boldsymbol{0}\\ \boldsymbol{F}_{3'}^e\end{Bmatrix}$$

可以看出，经过扩阶后得到的单元刚度阵仍是对称的，扩阶后的矩阵是扩阶单元刚度矩阵按单元的节点编号迭加得到。因此在程序实现时我们并不需要获得 $\mathbf{T}^e$，也不需要进行矩阵乘积运算。

（6.41）式关于所有单元求和，得

$$\boldsymbol{K}\boldsymbol{d}=\boldsymbol{F}\tag{6.42}$$

上式即为整个有限元系统节点位移和节点力间的平衡关系。$\boldsymbol{K}$ 为整体刚度矩阵或总刚阵。$\boldsymbol{F}$ 为有限元系统的体积力等效节点力和表面力等效节点力的和。由于单元之间相互作用的内力彼此抵消，因此式（6.42）中仅包含外部荷载所引起的等效节点力。由于组集后内力等效节点力总是可以彼此抵消的，今后的单元分析不再讨论内力等效节点力。

整体刚度矩阵具有很多重要的性质，说明如下。

一、整体刚度矩阵 $\boldsymbol{K}$ 中任一元素 K_{ij} 的物理意义是要使弹性体的第 j 个自由度发生单位位

移，而其他自由度都保持为零位移的状态下，第 i 个自由度上需要施加的外力；

二、整体刚度阵 $\boldsymbol{K}$ 的主对角元素总是正的；

三、整体刚度阵 $\boldsymbol{K}$ 是对称矩阵；

四、整体刚度阵 $\boldsymbol{K}$ 是一个稀疏阵；

五、整体刚度阵 $\boldsymbol{K}$ 是一个奇异阵、半正定阵，经约束刚体位移处理后，它是正定阵。

方程(6.42)在适当引入边界条件后，就能够解出所有未知节点的位移，进而可以求得单元应变、应力和节点应力。

6.3 收敛准则[1]

6.3.1 收敛准则

能否收敛于精确解是任何一种数值解法必须首先考虑的问题，有限单元法也是如此。我们总是希望随着网格的逐步细分，收敛到问题的精确解。从上面的分析可以看出，在单元形状确定后，位移模式的选择是关键。单元刚度矩阵和等效节点力的确定都依赖于位移模式的选择。因此，很难想象一个与真实位移解相差很远的位移模式可以得到一个很好的数值结果。已经证明，对于一个给定的位移模式，其刚度系数的数值要比实际大，因此，给定外部荷载时，有限元法求得的位移解比精确解小，随网格的细分，有限元位移解应能够逼近精确解。为保证有限元解的收敛性，位移模式应当满足如下条件：

一、位移模式必须包含单元的刚体位移

根据第三章的分析，刚体位移不会产生应变。单元位移模式不但要具有描述单元本身形变的能力，而且还要具有描述由于其他单元变形引起的单元刚体位移的能力。如图 6.5 所示平面有限元系统，假定各向同性材料的泊松比为零。显然单元⑤～⑧的应变和应力为零，由于单元①～④的压缩变形，⑤～⑧单元仍有一定的刚体位移。很难想象，不能描述刚体位移的有限单元能给出合理的计算结果。实际上，在一个有限元系统中，绝大多数单元都或多或少地具有刚体位移。还以图 6.5 所示系统为例，③和④单元也有刚体位移，当泊松比不等于零时，②单元也有刚体位移。

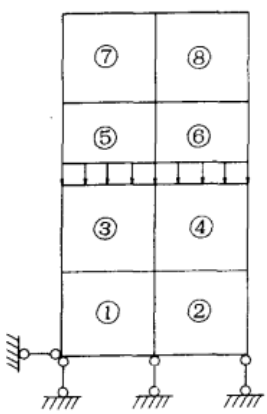


图 6.5 平面网格

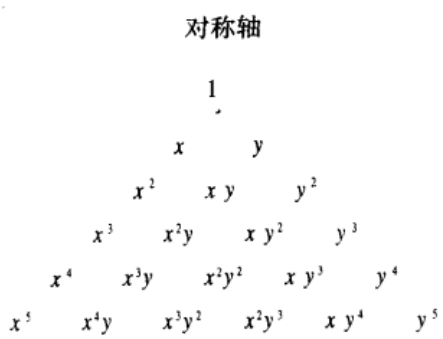


图 6.6 巴斯卡三角形

二、位移模式必须包含单元的常应变

每个单元的应变一般总可以分解两部分：一部分与单元坐标无关，另一部分与单元的坐标有关，前一部分称为常应变。从数学上看，当单元尺度无限缩小时，单元应变趋向于常应变，除非单元的位移模式中包含有常应变，否则就没有可能在有限元网格无限细分时收敛于正确解。

三、位移模式在单元内要连续、并使单元之间的位移协调

当选择多项式来构造位移模式时，单元内的位移连续性总是满足的。单元间的协调性要求单元之间不开裂不重叠。通常，当单元交界面上的位移取决于该交界面上的节点的位移时，可以保证位移的协调性。

在有限单元法中，满足条件一和条件二的单元称为完备单元；满足条件三的单元称为协调单元或保续单元。根据 3 节点平面三角形单元、4 节点矩形单元和四面体单元的插值形函数的性质可知，这三种类型的单元的位移模式都满足上述三个条件。

6.3.2　多项式位移模式阶次的选择

在选择多项式位移模式的阶次时，要考虑到解的收敛性，即要考虑位移模式的完备性和协调性。实践证明，完备性和协调性是十分重要的，但并不完全。选择位移模式阶次时，还应该考虑的另一个因素是，单元位移模式应当与单元局部坐标系的方位无关。这一性质称为几何各向同性。对于线性多项式，各向同性的要求通常等价于必须包含刚体位移和常应变状态。对于高项模式，就是不应有一个偏惠的坐标方向。经验验证，实现几何各向同性的一种方法是根据图 6.6 所示的巴斯卡三角形来选择完全的二维多项式的各项。在二维多项式中，若包含有三角形对称轴一边的任意一项，则必须同时包含另一边的对称项。

选择多项式模式要考虑的另一个因素，就是某自由度位移模式的多项式项数必须等于或稍大于与该自由度有关的单元节点自由度数，通常取等于。以我们熟悉的 2 节点杆单元和梁单元为例。2 节点杆单元内的位移模式是线性的，对于梁单元，由于转角反映了挠度沿长度方向的变化梯度，因此挠度的位移插值模式为三次的。

6.4　等参数单元

前面我们介绍了平面三角形单元，平面矩形单元和空间四面体单元，这些单元的特点是比较简单，为早期的有限元程序所采用。由于这些单元采用的是线性或双线性的位移模式，是对实际位移分布的最低级逼近，精度是受到限制的。矩形单元难以应用于非规则边界，因此我们希望构造出一些直边和曲边的四边形单元，以适应非规则的边界，同时具有较高的精度。

6.4.1　等参数杆单元

如图 6.7 所示的长为 2 的单元，坐标 ξ 的原点建立在杆的中间。现假定某物理量 q 在节点 1 和 2 处的值分别为 q_1 和 q_2，则该量在单元中任一点的值可线性插值得到

$$q(\xi) = N_1(\xi)q_1 + N_2(\xi)q_2 \tag{6.43}$$

式中，

$$N_1(\xi)=(1-\xi)/2 \tag{6.44}$$

$$N_2(\xi)=(1+\xi)/2 \tag{6.45}$$



图 6.7　2 节点杆单元

物理量 q 可以是位移，也可以是坐标，于是

$$\boldsymbol{u}=N_1(\xi)\boldsymbol{u}_1+N_2(\xi)\boldsymbol{u}_2 \tag{6.46}$$

$$\boldsymbol{x}=N_1(\xi)\boldsymbol{x}_1+N_2(\xi)\boldsymbol{x}_2 \tag{6.47}$$

式中，$\boldsymbol{u}_1$ 实际应当为 $\boldsymbol{u}_{1'}$ 代表单元节点 1 的位移矢量，而不是整体编号为 1 的节点的位移矢量。为简洁起见，这里略去代表单元节点局部编号的上标一撇，今后将不再重复说明。我们称 ξ 为自然坐标，N_i 为插值形函数，在自然坐标系中定义的长度为 2 的单元称为母单元。称这种位移模式和坐标变换式采用相同形函数的单元为等参数单元。

仔细分析(6.46)和(6.47)式是非常有益的。在前面介绍的三种单元中，单元的位移都是直接用整体坐标的多项式形式表示出来的。而上式则是通过一种新的途径将单元位移和整体坐标联系起来，那就是自然坐标 ξ 和定义在自然坐标中长度为 2 的母单元。只要母单元中的每一点都能和实际单元中的每一点构成一一对应关系，就能确保等参数单元的位移模式满足单元内位移连续的要求。等参单元的这种变换在数学上是十分简单、没有新意的，但它对有限单元法的发展具有十分重要的意义。

如果母单元上取 3 个节点，并且规定单元中间的节点位于母单元的中点，我们可以构造出高阶的插值形函数，

$$N_1=\frac{1-\xi}{2}-\frac{1}{2}(1-\xi^2)=\bar{N}_1-\frac{1}{2}N_3=\bar{N}_1-\bar{N}_1(\xi_3)N_3 \tag{6.48}$$

$$N_2=\frac{1+\xi}{2}-\frac{1}{2}(1-\xi^2)=\bar{N}_2-\frac{1}{2}N_3=\bar{N}_2-\bar{N}_2(\xi_3)N_3 \tag{6.49}$$

$$N_3=1-\xi^2 \tag{6.50}$$

式中，$\bar{N}_1$ 和 $\bar{N}_2$ 为 2 节点单元的插值函数，如式(6.44)和(6.45)所示，$\xi_3=0$ 为节点 3 的自然坐标。上式表明，以基本的线性插值函数为基础，将导致一个附加的插值函数对原有的函数进行修正。如果我们约定，节点 3 不存在时 $N_3=0$，则式(6.48)～(6.50)就是 2～3 节点杆单元通用的插值形函数。

下面给出计算 2～3 节点单元形函数及其导数的子程序。

```
      SUBROUTINE Shap1D( shape, coren, xloca, ndime, nnode,          &
                         lnode, xjaco, iflag )
!.....
!     PROGRAM
!           To calculate shape function and its derivations of one
```

```
!              local dimension.
!              if iflag == 1 derivations to local coordinate system.
!              if iflag == 2 derivations to global coordinate system.
!.....
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION coren( ndime, nnode ), lnode( nnode ), shape( 2, 3 )

      xjaco = 0.0D0
      shape( 1, 1 ) = -0.5D0
      shape( 1, 2 ) =  0.5D0
      shape( 2, 1 ) = ( 1.0D0 - xloca ) / 2.0D0
      shape( 2, 2 ) = ( 1.0D0 + xloca ) / 2.0D0
      IF( nnode .GE. 3 .AND. lnode( 3 ) .NE. 0 ) THEN
        shape( 1, 3 ) = -2.0D0 * xloca
        shape( 2, 3 ) =  1.0D0 - xloca * xloca
        shape( 1, 1 ) = shape( 1, 1 ) - 0.5D0 * shape( 1, 3 )
        shape( 1, 2 ) = shape( 1, 2 ) - 0.5D0 * shape( 1, 3 )
        shape( 2, 1 ) = shape( 2, 1 ) - 0.5D0 * shape( 2, 3 )
        shape( 2, 2 ) = shape( 2, 2 ) - 0.5D0 * shape( 2, 3 )
      END IF
      IF( iflag .EQ. 1 ) RETURN
      DO idime = 1, ndime
        fwork = 0.0D0
        DO inode = 1, nnode
          fwork = fwork + shape( 1, inode ) * coren( idime, inode )
        END DO
        xjaco = xjaco + fwork * fwork
      END DO
      xjaco = DSQRT( xjaco )
      DO inode = 1, nnode
        shape( 1, inode ) = shape( 1, inode ) / xjaco
      END DO
      RETURN
      END
```

6.4.2　4~8 节点平面等参数单元

将构造杆单元插值函数的方法推广到平面问题，可得到 4～8 节点平面等参单元。

一、4 节点平面等参元

如图 6.8 所示的在自然坐标系中边长为 2 的正方形单元，$O\xi\eta$ 为自然坐标系，原点在正方形的形心处。单元角节点 i 的自然坐标 (ξ_i,η_i) 分别为±1，因此单元四条边界的方程可用简单公式 $\xi\pm1=0$ 和 $\eta\pm1=0$ 逐一给出。

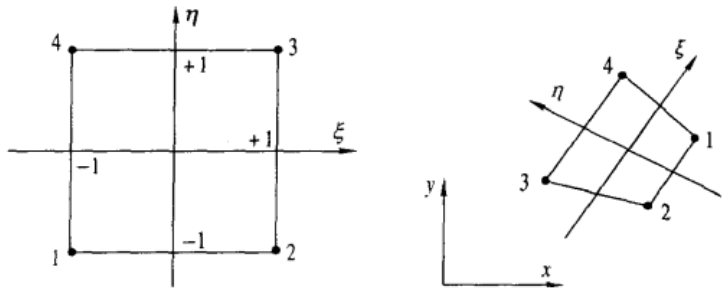


图 6.8 平面 4 节点单元

参照等参数杆单元，设单元位移模式和单元坐标为如下形式，

$$\boldsymbol{u}=\sum_{i=1}^{4}N_i\boldsymbol{u}_i=\boldsymbol{N}\boldsymbol{d}^e \tag{6.51}$$

$$\boldsymbol{x}=\sum_{i=1}^{4}N_i\boldsymbol{x}_i \tag{6.52}$$

$$\boldsymbol{N}=\left[N_1\boldsymbol{I}_{2\times2}\quad N_2\boldsymbol{I}_{2\times2}\quad N_3\boldsymbol{I}_{2\times2}\quad N_4\boldsymbol{I}_{2\times2}\right] \tag{6.53}$$

容易得到形函数 N_i 的表达式如下，

$$N_i=(1+\xi_i\xi)(1+\eta_i\eta)/4 \tag{6.54}$$

坐标变换式(6.52)使得图 6.8 中 $\xi\eta$ 平面上的四个角点分别映射成 xy 平面上的四个角点，它们的坐标是 $x_i, y_i(i=1,2,3,4)$。由于形函数是双线性的，$\xi\eta$ 平面上的正方形被映射到 xy 平面上以 x_i、y_i 为角点的四边形。显然，利用坐标变换式（6.52）可以把 xy 平面上的所有四边形单元，都映射到 $\xi\eta$ 平面上的正方形单元。下面我们来详细介绍这个单元的刚度矩阵和等效节点力的计算。

首先将位移的表达式（6.51）代入几何方程，便得应变分量的计算公式，

$$\boldsymbol{\varepsilon}=\boldsymbol{L}\boldsymbol{u}=\boldsymbol{B}\boldsymbol{d}^e=\left[\boldsymbol{B}_1\ \boldsymbol{B}_2\ \boldsymbol{B}_3\ \boldsymbol{B}_4\right]\boldsymbol{d}^e \tag{6.55}$$

式中，

$$\boldsymbol{d}^e=\left\{\boldsymbol{u}_1^T,\ \boldsymbol{u}_2^T,\ \boldsymbol{u}_3^T,\boldsymbol{u}_4^T\right\}^T\qquad \boldsymbol{u}_i=\{u_i,\ v_i\}^T \tag{6.56}$$

$$\boldsymbol{B}_i=\begin{bmatrix}N_{i,x} & 0\\ 0 & N_{i,y}\\ N_{i,y} & N_{i,x}\end{bmatrix}\qquad (i=1,2,3,4)$$

这里，记号 $N_{i,x}=\partial N_i/\partial x, N_{i,y}=\partial N_i/\partial y$。根据复合函数求导规则，这两个导数可以按下列步骤计算，首先

$$N_{i,\xi}=\frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi}+\frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi}$$

$$N_{i,\eta}=\frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta}+\frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta}$$

写成矩阵形式

$$\begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} \tag{6.57}$$

式中，

$$N_{i,\xi} = \partial N_i / \partial \xi \quad N_{i,\eta} = \partial N_i / \partial \eta,$$

$$x_\xi = \partial x / \partial \xi \quad x_\eta = \partial x / \partial \eta \quad y_\xi = \partial y / \partial \xi \quad y_\eta = \partial y / \partial \eta$$

矩阵

$$\boldsymbol{J} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} \tag{6.58}$$

就是坐标变换式（6.52）的雅可比矩阵，容易求得 $\boldsymbol{J}$ 的逆阵为

$$\boldsymbol{J}^{-1} = \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} y_\eta & -y_\xi \\ -x_\eta & x_\xi \end{bmatrix} \tag{6.59}$$

式中，$|\boldsymbol{J}|$ 为雅可比行列式，它用下式计算，

$$|\boldsymbol{J}| = x_\xi y_\eta - y_\xi x_\eta \tag{6.60}$$

其中

$$\begin{aligned} x_\xi &= \sum_{i=1}^{4} N_{i,\xi} x_i \qquad x_\eta = \sum_{i=1}^{4} N_{i,\eta} x_i \\ y_\xi &= \sum_{i=1}^{4} N_{i,\xi} y_i \qquad y_\eta = \sum_{i=1}^{4} N_{i,\eta} y_i \end{aligned} \tag{6.61}$$

当单元节点按逆时针方向编号时，雅可比行列式的值大于零，否则小于零。根据关系式（6.57）可以得到 $N_{i,x}$ 和 $N_{i,y}$ 的计算公式如下，

$$\begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} = \boldsymbol{J}^{-1} \begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} \tag{6.62}$$

利用物理关系，单元内的应力可以表示成，

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{d}^e \tag{6.63}$$

于是，单元刚度矩阵为

$$\boldsymbol{K}^e = \iint_{A^e} \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B} h \mathrm{d}x\mathrm{d}y = \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B} h |\boldsymbol{J}| \mathrm{d}\xi \mathrm{d}\eta \tag{6.64}$$

式中，A^e 为单元占据的面积，h 为单元的厚度。如果 h 为常数，可以将它从积分公式中提出。

采用类似的推导过程可以导出单元体积力等效节点力，

$$\boldsymbol{F}_f^e = \left\{ \boldsymbol{F}_{f1}^T \quad \boldsymbol{F}_{f2}^T \quad \boldsymbol{F}_{f3}^T \quad \boldsymbol{F}_{f4}^T \right\}^T \tag{6.65}$$

$$\boldsymbol{F}_{fi} = \int_{-1}^{1}\int_{-1}^{1} N_i \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} h |\boldsymbol{J}| \mathrm{d}\xi \mathrm{d}\eta \tag{6.66}$$

单元表面外力等效节点力为

$$\boldsymbol{F}_t^e = \left\{ \boldsymbol{F}_{t1}^T \quad \boldsymbol{F}_{t2}^T \quad \boldsymbol{F}_{t3}^T \quad \boldsymbol{F}_{t4}^T \right\}^T \tag{6.67}$$

$$\boldsymbol{F}_{ti} = \int_{\Gamma_t^e} N_i \begin{Bmatrix} \overline{t}_x \\ \overline{t}_y \end{Bmatrix} h \mathrm{d}\Gamma \tag{6.68}$$

式(6.64) 和(6.66)可以利用 Gauss 积分求得数值解。积分式(6.68)时，注意表面外力一般总是作用在单元的表面，因此，一定有 $\xi=\pm1$ 或 $\eta=\pm1$，则

$$d\Gamma=\sqrt{dx^2+dy^2}=\sqrt{x_\eta^2+y_\eta^2}d\eta \tag{6.69}$$

或

$$d\Gamma=\sqrt{dx^2+dy^2}=\sqrt{x_\xi^2+y_\xi^2}d\xi \tag{6.70}$$

这样式(6.68)也可以采用 Gauss 积分求其数值解。

可以证明，4 节点平面等参单元满足完备性和协调性条件。

二、4～8 节点平面等参元

在 4 节点平面等参数单元的每条边上增加一个节点就得到 8 节点等参数单元，如图 6.9 所示。这里的等参坐标变换，把边上的节点映射到母单元边界上时，总是使其成为对应边上的中点，因此常简称为边中节点。在有限元系统数据文件整理过程中，要特别注意单元节点的次序，本书采用的次序是：逆时针、先角节点后边中节点。

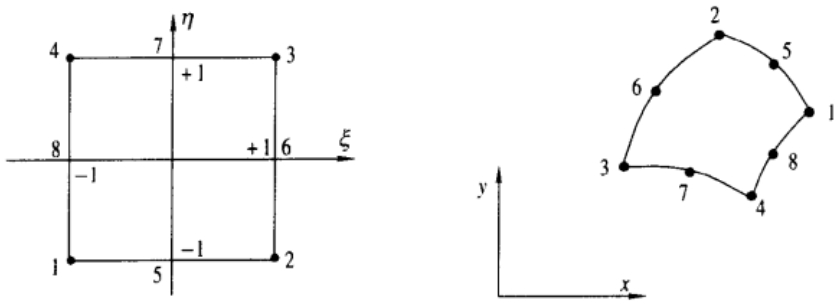


图 6.9 平面 8 节点单元

形函数可按如下方法构造，以 N_5 为例，在母单元中，使 N_5 在不含节点 5 的三条边上为零，并在节点 5 处等于 1，则

$$N_5=(\eta-1)(\xi^2-1)/2 \tag{6.71}$$

类似的有

$$N_6=(\xi+1)(1-\eta^2)/2 \tag{6.72}$$

$$N_7=(\eta+1)(1-\xi^2)/2 \tag{6.73}$$

$$N_8=(\xi-1)(\eta^2-1)/2 \tag{6.74}$$

角节点的形函数可写成如下形式，

$$N_i=\bar{N}_i-\sum_{j=1}^{4}\bar{N}_i(\xi_{j+4},\eta_{j+4})N_{j+4}\qquad(i=1,2,3,4) \tag{6.75}$$

式中，$\bar{N}_i$ 就是(6.54)式中的形函数，ξ_j、η_j $(j=i+4)$ 是节点 j 的自然坐标。

同 3 节点杆单元类似，这种线性组合的写法能方便地以统一的形式写出 4 节点、5 节点、6 节点、7 节点和 8 节点等一系列等参数单元形函数，例如对于没有节点 5 的 7 节点等参数单元，其形函数仍由（6.71）～（6.75）式给出，只需令 $N_5=0$ 即可。因此，形函数式（6.71）～（6.75），再加上一个约定，即某一边上边中节点不出现，就令其对应的形函数恒等于零，它们就是通常所称的 4~8 节点单元的形函数。这种 4~8 节点单元，在实际中是很有用的。

有了形函数之后，位移模式和坐标变换如下

$$\boldsymbol{u} = \sum_{i=1}^{8} N_i \boldsymbol{u}_i = \boldsymbol{N}\boldsymbol{d}^e \tag{6.76}$$

$$\boldsymbol{x} = \sum_{i=1}^{8} N_i \boldsymbol{x}_i \tag{6.77}$$

关于单元特性矩阵等的计算，可仿照 4 节点等参数单元的步骤进行。

如果在单元的每条边上有两个中间节点，就可构造出 12 节点等参数单元，其形函数也可按相同方法得到。在 4～8 节点单元中，要注意单元的节点数和单元实际节点数的区别，单元的节点数一般是指单元节点局部序号的最大值，不存在的节点应当用整体编号 0 代替。

下面给出计算 4～8 节点平面单元的形函数的子程序。

```
      SUBROUTINE Shap2D( shape, coren, xloca, yloca, lnode, xjaco,        &
                         mdime, ndime, nnode, ielem, iflag, nerrc )
!.....
!         P R O B L E M
!              To calculate shape functions and its' derivations.
!              if iflag == 1 derivations to local   coordinate sys.
!              if iflag == 2 derivations to grobal coordinate sys.
!.....
      IMPLICIT   DOUBLE PRECISION( a-h, o-z )
      DIMENSION                  fwork( 2, 3 )
      DIMENSION xloct( 4 ), xjaci( 2, 2 )
      DIMENSION yloct( 4 ), shape( 3, 8 )
      DIMENSION lnode( nnode ), coren( mdime, nnode )

      DATA xloct /-0.5D0,   0.5D0, 0.5D0, -0.5D0/
      DATA yloct /-0.5D0, -0.5D0, 0.5D0,   0.5D0/
!     计算 4 节点单元形函数及其对自然坐标的导数
      DO inode = 1, 4
        shape( 1, inode ) = ( 0.5D0 + yloct( inode ) * yloca ) *          &
                                      xloct( inode )
        shape( 2, inode ) = ( 0.5D0 + xloct( inode ) * xloca ) *          &
                                      yloct( inode )
        shape( 3, inode ) = ( 0.5D0 + xloct( inode ) * xloca ) *          &
                            ( 0.5D0 + yloct( inode ) * yloca )
      END DO
!     对所有边中节点的形函数设为零。
      DO inode = 5, 8
        DO idime = 1, 3
          shape( idime, inode ) = 0.0D0
        END DO
      END DO
!     开始计算边中节点形函数。
```

```
      xloc2 = ( 1.0D0 - xloca * xloca ) / 2.0D0
      yloc2 = ( 1.0D0 - yloca * yloca ) / 2.0D0
!     边中节点 5
      IF( nnode .GE. 5 .AND. lnode( 5 ) .NE. 0 ) THEN
        shape( 2, 5 ) = -xloc2
        shape( 1, 5 ) = -xloca * ( 1.0D0 - yloca )
        shape( 3, 5 ) =  xloc2 * ( 1.0D0 - yloca )
      END IF
!     边中节点 6
      IF( nnode .GE. 6 .AND. lnode( 6 ) .NE. 0 ) THEN
        shape( 1, 6 ) =  yloc2
        shape( 2, 6 ) = -yloca * ( 1.0D0 + xloca )
        shape( 3, 6 ) =  yloc2 * ( 1.0D0 + xloca )
      END IF
!     边中节点 7
      IF( nnode .GE. 7 .AND. lnode( 7 ) .NE. 0 ) THEN
        shape( 2, 7 ) =  xloc2
        shape( 1, 7 ) = -xloca * ( 1.0D0 + yloca )
        shape( 3, 7 ) =  xloc2 * ( 1.0D0 + yloca )
      END IF
!     边中节点 8
      IF( nnode .GE. 8 .AND. lnode( 8 ) .NE. 0 ) THEN
        shape( 1, 8 ) = -yloc2
        shape( 2, 8 ) = -yloca * ( 1.0D0 - xloca )
        shape( 3, 8 ) =  yloc2 * ( 1.0D0 - xloca )
      END IF
!     根据边中节点形函数值修正角节点形函数
      DO inode = 1, 4
        jnode = inode + 3
        knode = inode + 4
        IF( inode .EQ. 1 ) jnode = 8
        DO idime = 1, 3
          shape( idime, inode ) = shape( idime, inode ) - (                    &
          shape( idime, jnode ) + shape( idime, knode ) ) / 2.0D0
        END DO
      END DO

      IF( iflag .EQ. 1 ) RETURN
!     计算形函数对整体坐标的导数
      DO idime = 1, 2
        DO jdime = 1, ndime
          fwork( idime, jdime ) = 0.0D0
          DO inode = 1, nnode
            fwork( idime, jdime ) = fwork( idime, jdime ) +                    &
```

```
            shape( idime, inode ) * coren( jdime, inode )
          END DO
        END DO
      END DO

      IF( ndime .EQ. 2 ) THEN
        xjaco = fwork( 1, 1 ) * fwork( 2, 2 ) -                          &
                fwork( 1, 2 ) * fwork( 2, 1 )
        IF( xjaco .LT. 1.0D-8 ) THEN
          WRITE( 12, 2000 ) ielem
          nerrc = 158436
          RETURN
        END IF
        xjaci( 1, 1 ) =   fwork( 2, 2 ) / xjaco
        xjaci( 2, 2 ) =   fwork( 1, 1 ) / xjaco
        xjaci( 1, 2 ) = -fwork( 1, 2 ) / xjaco
        xjaci( 2, 1 ) = -fwork( 2, 1 ) / xjaco
        DO inode = 1, nnode
          dndgx = shape( 1, inode ) * xjaci( 1, 1 ) +                    &
                  shape( 2, inode ) * xjaci( 1, 2 )
          dndgy = shape( 1, inode ) * xjaci( 2, 1 ) +                    &
                  shape( 2, inode ) * xjaci( 2, 2 )
          shape( 1, inode ) = dndgx
          shape( 2, inode ) = dndgy
        END DO
      ELSE
        dadlx = fwork( 1, 2 ) * fwork( 2, 3 ) -                          &
                fwork( 1, 3 ) * fwork( 2, 2 )
        dadly = fwork( 1, 3 ) * fwork( 2, 1 ) -                          &
                fwork( 1, 1 ) * fwork( 2, 3 )
        dadlz = fwork( 1, 1 ) * fwork( 2, 2 ) -                          &
                fwork( 1, 2 ) * fwork( 2, 1 )
        xjaco = DSQRT( dadlx * dadlx + dadly * dadly +                       &
                       dadlz * dadlz )
      END IF
2000  FORMAT( 2x, '第', I5, '单元的雅可比行列式值为零或负值！' )
      RETURN
      END
```

6.4.3　空间轴对称等参数单元

在空间轴对称问题中，整体坐标系采用的是圆柱坐标，r 表示径向坐标，z 表示轴向坐标。采用与平面单元完全相同的 4～8 节点等参数单元，坐标变换式和位移模式分别具有下列形式，

$$\boldsymbol{x}=\sum_{i=1}^{n}N_i\boldsymbol{x}_i \tag{6.78}$$

$$\boldsymbol{u}=\sum_{i=1}^{n}N_i\boldsymbol{u}_i=\boldsymbol{N}\boldsymbol{d}^e \tag{6.79}$$

式中，n 是单元的节点数，

$$\boldsymbol{x}=\begin{Bmatrix}r\\z\end{Bmatrix}\qquad\qquad \boldsymbol{u}=\begin{Bmatrix}u\\w\end{Bmatrix}$$

应变计算公式是

$$\boldsymbol{\varepsilon}=\boldsymbol{L}\boldsymbol{u}=\boldsymbol{B}\,\boldsymbol{d}^e \tag{6.80}$$

式中，

$$\boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_1 & \boldsymbol{B}_2 & \cdots & \boldsymbol{B}_n\end{bmatrix}$$

$$\boldsymbol{B}_i=\begin{bmatrix}N_{i,r} & 0\\ 0 & N_{i,z}\\ N_{i,z} & N_{i,r}\\ 0 & N_i/r\end{bmatrix} \tag{6.81}$$

其中，$N_{i,r}$ 和 $N_{i,z}$ 分别表示 N_i 对 r 和 z 的偏导数，它们可按下式计算，

$$\begin{Bmatrix}N_{i,r}\\N_{i,z}\end{Bmatrix}=\boldsymbol{J}^{-1}\begin{Bmatrix}N_{i,\xi}\\N_{i,\eta}\end{Bmatrix} \tag{6.82}$$

雅可比矩阵 $\boldsymbol{J}$ 为

$$\boldsymbol{J}=\begin{bmatrix}r_\xi & z_\xi\\ r_\eta & z_\eta\end{bmatrix} \tag{6.83}$$

雅可比行列式为

$$|\boldsymbol{J}|=r_\xi z_\eta-r_\eta z_\xi \tag{6.84}$$

其中，

$$\begin{aligned}r_\xi&=\sum_{i=1}^{n}N_{i,\xi}r_i & r_\eta&=\sum_{i=1}^{n}N_{i,\eta}r_i\\ z_\xi&=\sum_{i=1}^{n}N_{i,\xi}z_i & z_\eta&=\sum_{i=1}^{n}N_{i,\eta}z_i\end{aligned} \tag{6.85}$$

应力计算公式是

$$\boldsymbol{\sigma}=\boldsymbol{D}\,\boldsymbol{B}\,\boldsymbol{d}^e \tag{6.86}$$

当 $r=0$ 时，即在对称轴上有 $\varepsilon_r=\varepsilon_\theta$，此时可以用 $N_{i,r}$ 代替 N_i/r 以消除式（6.81）中的奇异项。

单元刚度矩阵是

$$\boldsymbol{K}=2\pi\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{B}^T\,\boldsymbol{D}\,\boldsymbol{B}\,r|\boldsymbol{J}|\,\mathrm{d}\xi\mathrm{d}\eta \tag{6.87}$$

注意，有些文献给出的刚度矩阵没有系数 2π，它代表单位弧长的空间轴对称单元所具有的刚度。另外，不同于平面问题，空间轴对称问题实际存在的刚体自由度数最多只有一个，即沿

轴向的平动位移。

6.4.4　8～20 节点空间等参数单元

8～20 节点空间六面体等参数单元，可以由平面等参数单元推广得到。平面的 4 节点单元对应于空间的 8 节点单元，平面的 8 节点单元对应于空间的 20 节点单元，等等。空间等参数单元的母单元是边长为 2 的立方体，采用的自然坐标系是 $\xi\eta\zeta$，其原点在母单元的形心处。空间六面体等参数单元可以是直棱的六面体，也可以是曲面曲棱的六面体。图 6.10 是一个 20 节点等参数单元的示意图。

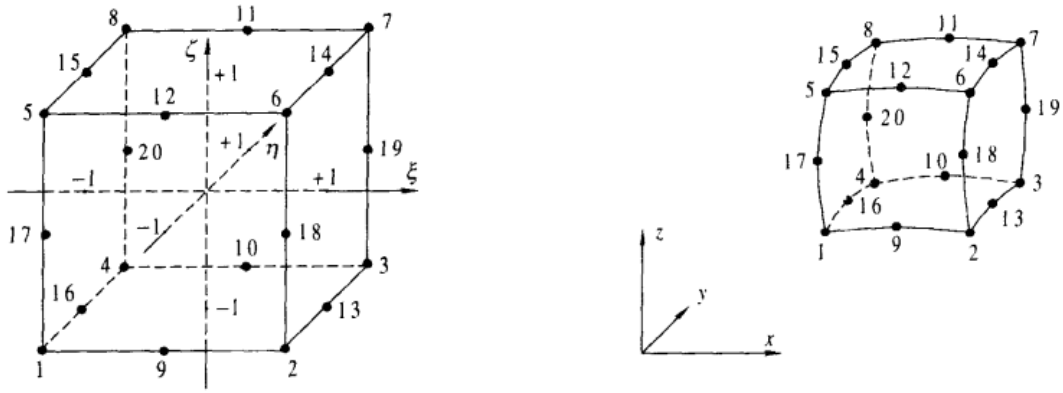


图 6.10　20 节点等参数单元

坐标变换式和位移模式可统一写成如下形式，

$$\boldsymbol{x}=\sum_{i=1}^{n}N_i\boldsymbol{x}_i \tag{6.88}$$

$$\boldsymbol{u}=\sum_{i=1}^{n}N_i\boldsymbol{u}_i=\boldsymbol{N}\boldsymbol{d}^e \tag{6.89}$$

式中，n 为单元的节点数，取 8～20。

$$\boldsymbol{x}=\{x,y,z\}^T \qquad \boldsymbol{u}=\{u,v,w\}^T$$

$$\boldsymbol{N}=\begin{bmatrix}N_1\boldsymbol{I}_{3\times3} & N_2\boldsymbol{I}_{3\times3} & \cdots & N_n\boldsymbol{I}_{3\times3}\end{bmatrix} \tag{6.90}$$

当 $n=8$ 时，即 8 节点等参数单元，首先来写出它的形函数

$$N_i=(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)/8 \quad (i=1,2,\cdots 8) \tag{6.91}$$

其中，ξ_i,η_i,ζ_i 是节点 i 的自然坐标，对于角节点它们分别为+1 或−1。

观察形函数式（6.91），其右端的每一项正好是三个平面方程左端（右端等于零）的乘积，节点 i 都不在这些平面内。将其他 7 个角节点坐标代入均有等于零的结果。将节点 i 的坐标 (ξ_i,η_i,ζ_i) 代入则正好等于 1，仿照这个办法，能写出节点 9 到 20 对应的形函数，

$$\begin{aligned}
N_i&=(1-\xi^2)(1+\eta_i\eta)(1+\zeta_i\zeta)/4 \quad (i=9,10,11,12)\\
N_i&=(1-\eta^2)(1+\xi_i\xi)(1+\zeta_i\zeta)/4 \quad (i=13,14,15,16)\\
N_i&=(1-\zeta^2)(1+\xi_i\xi)(1+\eta_i\eta)/4 \quad (i=17,18,19,20)
\end{aligned} \tag{6.92}$$

对于节点 9 到 20，ξ_i,η_i,ζ_i 分别取 0 或+1 或−1，例如 $\xi_9=0$，$\eta_9=-1$，$\zeta_9=-1$。

对于 8 节点等参数单元，有形函数式（6.91）。对于 20 节点等参数单元，边中节点形函数如式（6.92）所表示，这时角节点的形函数由如下的线性组合得出，

$$N_i = \bar{N}_i - \sum_{j=1}^{12} \bar{N}_i(\xi_{j+8}, \eta_{j+8}, \zeta_{j+8}) N_{j+8} \qquad (i = 1, 2, \cdots, 8) \tag{6.93}$$

其中，$\bar{N}_i$ 即为式（6.91）所表示的 8 节点等参单元的形函数。

类似地，如果增加一个约定：若某一个或某几个边中节点不存在，则在形函数（6.92）式中令相应的某一个形函数或某几个形函数恒等于零。有了这个约定，则（6.92）和式（6.93）就可作为 8~20 节点等参数单元的形函数。

根据几何关系式，单元应变计算公式是，

$$\boldsymbol{\varepsilon} = \boldsymbol{L}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{d}^e = [\boldsymbol{B}_1 \;\; \boldsymbol{B}_2 \;\; \cdots \;\; \boldsymbol{B}_n]\boldsymbol{d}^e \tag{6.94}$$

其中，

$$\boldsymbol{B}_i = \begin{bmatrix} N_{i,x} & 0 & 0 \\ 0 & N_{i,y} & 0 \\ 0 & 0 & N_{i,z} \\ 0 & N_{i,z} & N_{i,y} \\ N_{i,z} & 0 & N_{i,x} \\ N_{i,y} & N_{i,x} & 0 \end{bmatrix} \tag{6.95}$$

记号 $N_{i,x}$、$N_{i,y}$ 和 $N_{i,z}$ 分别表示 N_i 对 x、y 和 z 的偏导数。根据复合函数求导数的规则，它们与 $N_{i,\xi}$、$N_{i,\eta}$ 和 $N_{i,\zeta}$ 有如下的关系式

$$\begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \\ N_{i,\zeta} \end{Bmatrix} = \begin{bmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & x_\zeta \end{bmatrix} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{Bmatrix} = \boldsymbol{J} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{Bmatrix} \tag{6.96}$$

其中，

$$x_\xi = \sum_{i=1}^{n} N_{i,\xi} x_i \qquad y_\xi = \sum_{i=1}^{n} N_{i,\xi} y_i \qquad z_\xi = \sum_{i=1}^{n} N_{i,\xi} z_i \tag{6.97}$$

而 $N_{i,\xi}$ 等可以通过对式（6.92）和式（6.93）求导数得出。其他各量可同样求出。由式（6.97）可得

$$\begin{Bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{Bmatrix} = \boldsymbol{J}^{-1} \begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \\ N_{i,\zeta} \end{Bmatrix} \tag{6.98}$$

应力的计算公式是

$$\boldsymbol{\sigma} = \boldsymbol{D}\,\boldsymbol{B}\,\boldsymbol{d}^e \tag{6.99}$$

单元刚度矩阵为

$$\boldsymbol{K}^e = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}^T\,\boldsymbol{D}\,\boldsymbol{B}\,|\boldsymbol{J}|\,\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{6.100}$$

体积力等效节点力和表面力等效节点力可同平面问题一样给出。

下面给出计算 8～20 节点六面体单元形函数及其导数的子程序。

```
      SUBROUTINE Shap3D( xloca, yloca, zloca, coren, shape, xjaco,        &
                         ndime, nnode, lnode, ielem, iflag, nerrc )
!.....
!     P R O G R A M
!          To set shape function of three dimensional elements with
!          8 to 21 nodes.
!.....
      IMPLICIT DOUBLE PRECISION ( a-h, o-z )
      DIMENSION xjaxi( 3,   3 ), xloct( 8 )
      DIMENSION xjaxm( 3,   3 ), yloct( 8 ), lnode( nnode )
      DIMENSION shape( 4, 21 ), zloct( 8 ), coren( ndime, nnode )
      data xloct/-1., 1., 1.,-1.,-1., 1., 1.,-1./
      data yloct/-1.,-1., 1., 1.,-1.,-1., 1., 1./
      data zloct/-1.,-1.,-1.,-1., 1., 1., 1., 1./
!.....
!     8 节点线性形函数及其对自然坐标的导数
!.....
      DO inode = 1, 21
       DO idime = 1, 4
        shape( idime, inode ) = 0.0D0
       END DO
      END DO
      IF( nnode .lt. 8 ) RETURN
      DO inode = 1, 8
       xtemp = 1.0D0 + xloct( inode ) * xloca
       ytemp = 1.0D0 + yloct( inode ) * yloca
       ztemp = 1.0D0 + zloct( inode ) * zloca
       shape( 4, inode ) = 0.125D0 * xtemp * ytemp * ztemp
       shape( 1, inode ) = 0.125D0 * ytemp * ztemp * xloct( inode )
       shape( 2, inode ) = 0.125D0 * ztemp * xtemp * yloct( inode )
       shape( 3, inode ) = 0.125D0 * xtemp * ytemp * zloct( inode )
      END DO
!.....
!     边中节点形函数及其对自然坐标的导数
!.....
      xlocm =   1.0D0 - xloca
      ylocm =   1.0D0 - yloca
      zlocm =   1.0D0 - zloca
      xlocp =   1.0D0 + xloca
      ylocp =   1.0D0 + yloca
      zlocp =   1.0D0 + zloca
      xtemp = ( 1.0D0 - xloca * xloca ) / 4.
      ytemp = ( 1.0D0 - yloca * yloca ) / 4.
      ztemp = ( 1.0D0 - zloca * zloca ) / 4.
```

```
IF( nnode .GE. 9 .AND. lnode( 9 ) .NE. 0 ) THEN
 shape( 4, 9 ) = ylocm * zlocm * xtemp
 shape( 1, 9 ) =-0.5D0 * xloca * ylocm * zlocm
 shape( 2, 9 ) =-xtemp * zlocm
 shape( 3, 9 ) =-xtemp * ylocm
END IF
IF( nnode .GE. 10 .AND. lnode( 10 ) .NE. 0 ) THEN
 shape( 4, 10 ) = ylocp * zlocm * xtemp
 shape( 1, 10 ) =-0.5D0 * xloca * ylocp * zlocm
 shape( 2, 10 ) = xtemp * zlocm
 shape( 3, 10 ) =-xtemp * ylocp
END IF
IF( nnode .GE. 11 .AND. lnode( 11 ) .NE. 0 ) THEN
 shape( 4, 11 ) = ylocp * zlocp * xtemp
 shape( 1, 11 ) =-0.5D0 * xloca * ylocp * zlocp
 shape( 2, 11 ) = xtemp * zlocp
 shape( 3, 11 ) = xtemp * ylocp
END IF
IF( nnode .GE. 12 .AND. lnode( 12 ) .NE. 0 ) THEN
 shape( 4, 12 ) = ylocm * zlocp * xtemp
 shape( 1, 12 ) =-0.5D0 * xloca * ylocm * zlocp
 shape( 2, 12 ) =-xtemp * zlocp
 shape( 3, 12 ) = xtemp * ylocm
END IF
IF( nnode .GE. 13 .AND. lnode( 13 ) .NE. 0 ) THEN
 shape( 4, 13 ) = zlocm * xlocp * ytemp
 shape( 1, 13 ) = ytemp * zlocm
 shape( 2, 13 ) =-0.5D0 * yloca * zlocm *xlocp
 shape( 3, 13 ) =-ytemp * xlocp
END IF
IF( nnode .GE. 14 .AND. lnode( 14 ) .NE. 0 ) THEN
 shape( 4, 14 ) = xlocp * zlocp * ytemp
 shape( 1, 14 ) = ytemp * zlocp
 shape( 2, 14 ) =-0.5D0 * yloca * xlocp * zlocp
 shape( 3, 14 ) = ytemp * xlocp
END IF
IF( nnode .GE. 15 .AND. lnode( 15 ) .NE. 0 ) THEN
 shape( 4, 15 ) = xlocm * zlocp * ytemp
 shape( 1, 15 ) =-zlocp * ytemp
 shape( 2, 15 ) =-0.5D0 * yloca * xlocm * zlocp
 shape( 3, 15 ) = ytemp * xlocm
END IF
IF( nnode .GE. 16 .AND. lnode( 16 ) .NE. 0 ) THEN
 shape( 4, 16 ) = xlocm * zlocm * ytemp
 shape( 1, 16 ) =-zlocm * ytemp
```

```
      shape( 2, 16 ) =-0.5D0 * yloca * xlocm * zlocm
      shape( 3, 16 ) =-xlocm * ytemp
     END IF
     IF( nnode .GE. 17 .AND. lnode( 17 ) .NE. 0 ) THEN
      shape( 4, 17 ) = ylocm * xlocm * ztemp
      shape( 1, 17 ) =-ylocm * ztemp
      shape( 2, 17 ) =-xlocm * ztemp
      shape( 3, 17 ) =-0.5D0 * zloca * xlocm * ylocm
     END IF
     IF( nnode .GE. 18 .AND. lnode( 18 ) .NE. 0 ) THEN
      shape( 4, 18 ) = xlocp * ylocm * ztemp
      shape( 1, 18 ) = ylocm * ztemp
      shape( 2, 18 ) =-xlocp * ztemp
      shape( 3, 18 ) =-0.5D0 * zloca * xlocp * ylocm
     END IF
     IF( nnode .GE. 19 .AND. lnode( 19 ) .NE. 0 ) THEN
      shape( 4, 19 ) = xlocp * ylocp * ztemp
      shape( 1, 19 ) = ylocp * ztemp
      shape( 2, 19 ) = xlocp * ztemp
      shape( 3, 19 ) =-0.5D0 * zloca * xlocp * ylocp
     END IF
     IF( nnode .GE. 20 .AND. lnode( 20 ) .NE. 0 ) THEN
      shape( 4, 20 ) = xlocm * ylocp * ztemp
      shape( 1, 20 ) =-ylocp * ztemp
      shape( 2, 20 ) = xlocm * ztemp
      shape( 3, 20 ) =-0.5D0 * zloca * xlocm * ylocp
     END IF
     IF( nnode .GE. 21 .AND. lnode( 21 ) .NE. 0 ) THEN
      shape( 4, 21 ) = 64. * xtemp * ytemp * ztemp
      shape( 1, 21 ) =-32. * xloca * ytemp * ztemp
      shape( 2, 21 ) =-32. * yloca * xtemp * ztemp
      shape( 3, 21 ) =-32. * xtemp * ytemp * zloca
!     边中节点形函数对角节点的修正
      DO idime = 1, 4
       DO inode = 1, 8
        shape( idime, inode ) = shape( idime, inode ) -                    &
                                0.25D0 * shape( idime,      21 )
       END DO
       DO inode = 9, 20
        IF( nnode .GE. 9 .AND. lnode( inode ) .NE. 0 )                      &
         shape( idime, inode ) = shape( idime, inode ) -                    &
                                 0.5D0 * shape( idime,      21 )
       END DO
      END DO
     END IF
```

```
      DO idime = 1, 4
       shape( idime,   1 ) = shape( idime,   1 ) - 0.5D0 * (                    &
       shape( idime,   9 ) + shape( idime, 16 ) + shape( idime, 17 ) )
       shape( idime,   2 ) = shape( idime,   2 ) - 0.5D0 * (                    &
       shape( idime,   9 ) + shape( idime, 18 ) + shape( idime, 13 ) )
       shape( idime,   3 ) = shape( idime,   3 ) - 0.5D0 * (                    &
       shape( idime, 13 ) + shape( idime, 19 ) + shape( idime, 10 ) )
       shape( idime,   4 ) = shape( idime,   4 ) - 0.5D0 * (                    &
       shape( idime, 10 ) + shape( idime, 16 ) + shape( idime, 20 ) )
       shape( idime,   5 ) = shape( idime,   5 ) - 0.5D0 * (                    &
       shape( idime, 12 ) + shape( idime, 15 ) + shape( idime, 17 ) )
       shape( idime,   6 ) = shape( idime,   6 ) - 0.5D0 * (                    &
       shape( idime, 12 ) + shape( idime, 14 ) + shape( idime, 18 ) )
       shape( idime,   7 ) = shape( idime,   7 ) - 0.5D0 * (                    &
       shape( idime, 11 ) + shape( idime, 14 ) + shape( idime, 19 ) )
       shape( idime,   8 ) = shape( idime,   8 ) - 0.5D0 * (                    &
       shape( idime, 11 ) + shape( idime, 15 ) + shape( idime, 20 ) )
      END DO
      DO idime = 1, ndime
       DO jdime = 1, 3
        ftemp = 0.0D0
        DO inode = 1, nnode
         ftemp = ftemp + coren( idime, inode ) *                                &
                         shape( jdime, inode )
        END DO
        xjaxm( jdime, idime ) = ftemp
       END DO
      END DO
      xjaxi( 1, 1 ) = xjaxm( 2, 2 ) * xjaxm( 3, 3 ) -                          &
                      xjaxm( 2, 3 ) * xjaxm( 3, 2 )
      xjaxi( 2, 1 ) = xjaxm( 2, 3 ) * xjaxm( 3, 1 ) -                          &
                      xjaxm( 2, 1 ) * xjaxm( 3, 3 )
      xjaxi( 3, 1 ) = xjaxm( 2, 1 ) * xjaxm( 3, 2 ) -                          &
                      xjaxm( 2, 2 ) * xjaxm( 3, 1 )
      xjaco = 0.0D0
      DO idime = 1, 3
       xjaco = xjaco + xjaxm( 1, idime ) * xjaxi( idime, 1 )
      END DO
      IF( iflag .EQ. 1 ) RETURN
!     计算形函数对整体坐标的导数
      xjaxi( 1, 2 ) = xjaxm( 3, 2 ) * xjaxm( 1, 3 ) -                          &
                      xjaxm( 3, 3 ) * xjaxm( 1, 2 )
      xjaxi( 2, 2 ) = xjaxm( 3, 3 ) * xjaxm( 1, 1 ) -                          &
                      xjaxm( 3, 1 ) * xjaxm( 1, 3 )
      xjaxi( 3, 2 ) = xjaxm( 3, 1 ) * xjaxm( 1, 2 ) -                          &
```

```
                    xjaxm( 3, 2 ) * xjaxm( 1, 1 )
      xjaxi( 1, 3 ) = xjaxm( 1, 2 ) * xjaxm( 2, 3 ) -                    &
                    xjaxm( 1, 3 ) * xjaxm( 2, 2 )
      xjaxi( 2, 3 ) = xjaxm( 1, 3 ) * xjaxm( 2, 1 ) -                    &
                    xjaxm( 1, 1 ) * xjaxm( 2, 3 )
      xjaxi( 3, 3 ) = xjaxm( 1, 1 ) * xjaxm( 2, 2 ) -                    &
                    xjaxm( 1, 2 ) * xjaxm( 2, 1 )
      IF( xjaco .LE. 1.0D-8 ) THEN
        WRITE( 12, 2000 ) ielem
        nerrc = 158436
        RETURN
      END IF
      DO idime = 1, 3
       DO jdime = 1, 3
        xjaxi( idime, jdime ) = xjaxi( idime, jdime ) / xjaco
       END DO
      END DO

      DO inode = 1, nnode
       xtemp = 0.0D0
       ytemp = 0.0D0
       ztemp = 0.0D0
       DO idime = 1, 3
        xtemp = xtemp + xjaxi( 1, idime ) * shape( idime, inode )
        ytemp = ytemp + xjaxi( 2, idime ) * shape( idime, inode )
        ztemp = ztemp + xjaxi( 3, idime ) * shape( idime, inode )
       END DO
       shape( 1, inode ) = xtemp
       shape( 2, inode ) = ytemp
       shape( 3, inode ) = ztemp
      END DO
2000   FORMAT( 2x, '第', I5, '单元的雅可比行列式值为零或负值！' )
      RETURN
      END
```

6.5　质量阵、阻尼阵和温度应力

6.5.1　一致质量阵和集中质量阵

对于动力学问题，物体运动过程中的惯性项不可忽略。根据达朗贝尔原理，将加速度引起的惯性力 $-\rho\ddot{\boldsymbol{u}}$ 看成是体积力的一个组成部分，将动力学问题等效为静力学问题。下面我们

来计算单元惯性力 $-\rho\ddot{\boldsymbol{u}}$ 引起的等效节点力 $\boldsymbol{F}_{\mathrm{I}}^{e}$

$$\boldsymbol{F}_{\mathrm{I}}^{e}=-\int_{V^e}\boldsymbol{N}^{T}\rho\ddot{\boldsymbol{u}}\mathrm{d}V=-\int_{V^e}\boldsymbol{N}^{T}\boldsymbol{N}\rho\mathrm{d}V\ddot{\boldsymbol{d}}^{e}\equiv-\boldsymbol{M}^{e}\ddot{\boldsymbol{d}}^{e} \tag{6.101}$$

式中，

$$\boldsymbol{M}^{e}=\int_{V^e}\boldsymbol{N}^{T}\boldsymbol{N}\rho\mathrm{d}V \tag{6.102}$$

称 $\boldsymbol{M}^{e}$ 为单元一致质量阵。由此可以得到以节点位移表示的单元运动方程，

$$\boldsymbol{M}^{e}\ddot{\boldsymbol{d}}^{e}+\boldsymbol{K}^{e}\boldsymbol{d}^{e}=\boldsymbol{F}_{f}^{e}+\boldsymbol{F}_{t}^{e}+\hat{\boldsymbol{F}}_{t}^{e} \tag{6.103}$$

利用选择矩阵 $\boldsymbol{T}^{e}$ 扩阶后组集，得

$$\boldsymbol{M}\ddot{\boldsymbol{d}}+\boldsymbol{K}\boldsymbol{d}=\boldsymbol{F} \tag{6.104}$$

此式即为以节点位移表示的有限元系统的运动方程，$\boldsymbol{M}$ 为总体一致质量阵。总体一致质量阵一般总是正定的。一致质量阵是根据能量来确定的，我们也可以按照一定的分配法则将单元的质量分配到单元的各个节点上去。例如，对于三维问题我们可以把单元质量矩阵写成如下形式，

$$\boldsymbol{M}=\mathrm{diag}(m_1,m_1,m_1,m_2,m_2,m_2,\cdots m_n,m_n,m_n) \tag{6.105}$$

式中，

$$m_k=\int_{V^e}N_k\rho\mathrm{d}V \tag{6.106}$$

称 (6.105) 式给出的对角质量阵为集中质量阵。显然，集中质量阵较一致质量阵更为直观，但其理论推导不够严密。

6.5.2 阻尼阵

物质运动或振动过程中，或多或少都存在一定的阻尼，产生一定的阻尼力。影响结构阻尼的因素十分复杂，但总的来说可以分为两大部分：一部分是由于材料本身的粘滞性引起的；另一部分是由于支座、空气阻力等因素引起。对于材料阻尼，我们通常可将材料的本构关系表示为如下 Maxwell 形式，

$$\boldsymbol{\sigma}=\boldsymbol{D}\boldsymbol{\varepsilon}+\boldsymbol{\eta}\dot{\boldsymbol{\varepsilon}} \tag{6.107}$$

阻尼系数矩阵 $\boldsymbol{\eta}$ 的确定比较困难，常引入一定假设，如 $\boldsymbol{\eta}=\beta\boldsymbol{D}$，也有假设体积变形与粘性变形无关等，此时单元内力所做的虚功可以表示为

$$\int_{V^e}(\delta\boldsymbol{\varepsilon})^{T}\boldsymbol{\sigma}\mathrm{d}V=\left(\delta\boldsymbol{d}^{e}\right)^{T}\left[\boldsymbol{K}^{e}\boldsymbol{d}^{e}+\boldsymbol{C}_{M}^{e}\dot{\boldsymbol{d}}^{e}\right] \tag{6.108}$$

式中，

$$\boldsymbol{C}_{M}^{e}=\int_{V^e}\boldsymbol{B}^{T}\boldsymbol{\eta}\boldsymbol{B}\mathrm{d}V \tag{6.109}$$

当 $\boldsymbol{\eta}=\beta\boldsymbol{D}$ 时，

$$\boldsymbol{C}_{M}^{e}=\beta\boldsymbol{K}^{e} \tag{6.110}$$

另一部分阻尼可假定与速度成正比，记比例系数为 μ。类似一致质量阵的推导，这部分阻尼引起的单元等效节点力为

$$\boldsymbol{F}_{B}^{e}=-\boldsymbol{C}_{B}^{e}\cdot\dot{\boldsymbol{d}}^{e} \tag{6.111}$$

$$\boldsymbol{C}_B^e = \int_{V^e} \boldsymbol{N}^T \boldsymbol{N} \cdot \mu \mathrm{d}V \tag{6.112}$$

如果比例系数 μ 和质量密度 ρ 的比值在整个单元中为常数，记为 α。比较(6.102) 和(6.112)式可知，阻尼阵和一致质量阵成比例，

$$\boldsymbol{C}_B^e = \alpha \boldsymbol{M}^e \tag{6.113}$$

综合以上各式可得

$$\boldsymbol{M}^e \ddot{\boldsymbol{d}}^e + \boldsymbol{C}^e \dot{\boldsymbol{d}}^e + \boldsymbol{K}^e \boldsymbol{d}^e = \boldsymbol{F}_f^e + \boldsymbol{F}_t^e + \hat{\boldsymbol{F}}_t^e \tag{6.114}$$

组集后得

$$\boldsymbol{M}\ddot{\boldsymbol{d}} + \boldsymbol{C}\dot{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{d} = \boldsymbol{F} \tag{6.115}$$

其中，

$$\boldsymbol{C} = \boldsymbol{C}_M + \boldsymbol{C}_B \tag{6.116}$$

由于影响结构阻尼的因素十分复杂，工程中常常采用 Rayleigh 阻尼，即假设总体阻尼阵是满足如下形式的比例阻尼矩阵，

$$\boldsymbol{C} = \alpha \boldsymbol{M} + \beta \boldsymbol{K} \tag{6.117}$$

其中 α 和 β 为两个系数，由实验确定。

6.5.3 温度应力

如果考虑温度变化对应力场的影响，应力应变关系可表示为

$$\boldsymbol{\sigma} = \boldsymbol{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_T) \tag{6.118}$$

式中，$\boldsymbol{\varepsilon}_T$ 是温度变化引起的应变。以空间问题为例，$\boldsymbol{\varepsilon}_T$ 可以表示为

$$\boldsymbol{\varepsilon}_T = \alpha \boldsymbol{m} \Delta T \tag{6.119}$$

其中，α 为热膨胀系数，$\boldsymbol{m} = \mathrm{diag}(1,1,1,0,0,0)$，$\Delta T$ 为相对参考温度的温升。单元中任意一点的温度变化可以采用位移插值的方法由节点插值得到，

$$\Delta T = \sum_{i=1}^{n} N_i \Delta T_i \tag{6.120}$$

代入虚功方程的左端，

$$\int_{V^e} (\delta\boldsymbol{\varepsilon})^T \boldsymbol{\sigma} \mathrm{d}V = (\delta \boldsymbol{d}^e)^T \left(\int_{V^e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \mathrm{d}V \, \boldsymbol{d}^e - \int_{V^e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{\varepsilon}_T \mathrm{d}V \right) \tag{6.121}$$

由上式不难看出由温度变化引起的等效节点力为

$$\boldsymbol{F}_T^e = \int_{V^e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{\varepsilon}_T \mathrm{d}V \tag{6.122}$$

6.6 节点应力

位移法有限元分析中首先求得节点位移，然后由几何方程和本构关系求得应力，所以位移的精度高于应力。沿着单元的边界，位移是连续的，但位移的导数往往是不连续的，因此在单元边界上应力往往是不连续的，如图 6.11 所示。经验表明，积分点上算得的应力具有最高的精度，而在节点上应力的计算值精度最低。

然而实际工程中感兴趣的是边缘和节点上的应力，为了克服边缘和节点上应力不连续和精度较差等缺点，对应力的计算结果需要进行整理。常用的处理方法有绕节点平均法、两节点平均法和应力修匀法等。绕节点平均法和两节点平均法主要用于普通的低阶单元应力处理。对于等参单元来说，应力修匀法是比较有效的节点应力计算方法。下面介绍应力修匀法的求解过程。



图 6.11 修匀与未修匀应力分布图

由于所有应力分量的应力修匀过程是相同的，本节以 σ 代表应力矢量中的任意一个分量。设修匀后单元节点的应力为 $\hat{\sigma}_1,\hat{\sigma}_2,\ldots,\hat{\sigma}_p$，单元内任意一点修匀后的应力可用形函数插值表示如下，

$$\hat{\sigma}=\sum_{k=1}^{p}\hat{N}_k\hat{\sigma}_k \tag{6.123}$$

其中，p 为单元修匀应力节点数，不大于单元的非零节点数，$\hat{N}_k$ 为单元应力插值形函数。考虑到应力阶次比位移阶次低，可以取 p 小于单元的非零节点数。例如，4～8 节点平面等参单元，通常取四个角节点进行应力修匀计算，也就是 $p=4$，此时，应力插值形函数比位移插值形函数低一阶。又如 8～20 节点的六面体等参单元，也可取八个角节点进行应力修匀计算，也就是 $p=8$。所有边中节点的修匀应力值等于与它共边的角节点修匀应力的平均值。

单元内任意点修匀应力和未修匀应力 σ 之间的偏差为

$$\delta=\sigma-\hat{\sigma} \tag{6.124}$$

修匀的应力应该使得 δ 在单元内尽可能小。

由于单元高斯点应力计算值比较准确，如果偏差 δ 在高斯点处很小，则可以认为修匀后的应力是合理的。记第 i 个高斯点处的偏差为 δ^i，根据(6.123)和(6.124)式有

$$\delta^i=\sigma^i-\sum_{k=1}^{p}\hat{N}_k^i\hat{\sigma}_k \qquad (i=1,2,\cdots,NG) \tag{6.125}$$

式中，上标 i 代表在第 i 个高斯点处取值，NG 为高斯点数。我们的目标是使得

$$\delta^i=0 \qquad (i=1,2,\cdots,NG) \tag{6.126}$$

当 $p>NG$ 时，方程数少而待求的物理量多，解不惟一。因此，高斯点数必须不小于单元修匀应力节点数。

当 $p=NG$ 时，式(6.126)代表 NG 个线性方程，解之可得节点修匀应力 $\hat{\sigma}_k$。

当 $p>NG$ 时，式(6.126)没有满足条件的精确解，可采用最小二乘法求最佳近似解。根据最小二乘法，有

$$\boldsymbol{A}^T\boldsymbol{A}\hat{\boldsymbol{\sigma}}=\boldsymbol{A}^T\boldsymbol{\sigma}^G \tag{6.127}$$

式中，

$$\hat{\boldsymbol{\sigma}}=\left\{\hat{\sigma}_1 \quad \hat{\sigma}_2 \quad \cdots \quad \hat{\sigma}_p\right\}^T$$

$$\boldsymbol{\sigma}^G=\left\{\sigma^1 \quad \sigma^2 \quad \cdots \quad \sigma^{NG}\right\}^T$$

$$\boldsymbol{A}=\begin{bmatrix} \hat{N}_1^1 & \hat{N}_2^1 & \cdots & \hat{N}_p^1 \\ \hat{N}_1^2 & \hat{N}_2^2 & \cdots & \hat{N}_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{N}_1^{NG} & \hat{N}_2^{NG} & \cdots & \hat{N}_p^{NG} \end{bmatrix}$$

求解(6.127)式可得修匀节点应力。

对于 4～8 节点平面等参单元，以四个角节点的修匀应力为基本未知量，即 $p=4$，采用 2×2 的高斯积分点，由(6.126)式可得如下方程组，

$$\begin{Bmatrix} \hat{\sigma}_1 \\ \hat{\sigma}_2 \\ \hat{\sigma}_3 \\ \hat{\sigma}_4 \end{Bmatrix}=\begin{bmatrix} a & b & c & b \\ b & a & b & c \\ c & b & a & b \\ b & c & b & a \end{bmatrix}\begin{Bmatrix} \sigma^1 \\ \sigma^2 \\ \sigma^3 \\ \sigma^4 \end{Bmatrix} \tag{6.128}$$

式中，

$$a=1+\frac{\sqrt{3}}{2} \qquad b=-\frac{1}{2} \qquad c=1-\frac{\sqrt{3}}{2}$$

对于 8～20 节点空间等参单元，以八个角节点的修匀应力为基本未知量，即 $p=8$，采用 2×2×2 的高斯积分点，由(6.126)式可得如下方程组，

$$\begin{Bmatrix} \hat{\sigma}_1 \\ \hat{\sigma}_2 \\ \hat{\sigma}_3 \\ \hat{\sigma}_4 \\ \hat{\sigma}_5 \\ \hat{\sigma}_6 \\ \hat{\sigma}_7 \\ \hat{\sigma}_8 \end{Bmatrix}=\begin{bmatrix} a & b & c & b & b & c & d & c \\ b & a & b & c & c & b & c & d \\ c & b & a & b & d & c & b & c \\ b & c & b & a & c & d & c & b \\ b & c & d & c & a & b & c & b \\ c & b & c & d & b & a & b & c \\ d & c & b & c & c & b & a & b \\ c & d & c & b & b & c & b & a \end{bmatrix}\begin{Bmatrix} \sigma^1 \\ \sigma^2 \\ \sigma^3 \\ \sigma^4 \\ \sigma^5 \\ \sigma^6 \\ \sigma^7 \\ \sigma^8 \end{Bmatrix} \tag{6.129}$$

式中，

$$a=\frac{5+3\sqrt{3}}{4} \qquad b=-\frac{\sqrt{3}+1}{4} \qquad c=\frac{\sqrt{3}-1}{4} \qquad d=\frac{5-3\sqrt{3}}{4}$$

所有边中节点的修匀应力由共边角节点的修匀应力取平均值得到。单元内任意点的修匀应力由角节点修匀应力插值得到。

按上述方法求得单元节点的修匀应力后，相邻单元在同一个节点处得到的修匀应力可能不同，还需要一些其他平均方法整理，如绕节点单元算术平均或加权平均。

6.7 线性有限单元法的实施步骤与注意事项

根据以上分析，有限单元法解决力学问题的具体步骤概括如下：

(1) 确定问题的类型和选用的单元类型以及相应的边界条件；

(2) 将要计算的结构进行网格划分，并对节点和单元进行整体编号；

(3) 列出节点坐标、单元各节点的总体编号，载荷信息和约束信息作为输入信息；

(4) 依次对所有单元循环，计算单元的刚度矩阵和单元等效节点力，组集形成总体刚度和总体等效节点力矢量；

(5) 处理约束及消除刚体位移；

(6) 求解线性方程组，求得各节点位移；

(7) 计算单元应力和节点应力；

(8) 结果提取和分析。

通常步骤（4）~（7）均由有限元分析软件完成，步骤（1）~（3）由人工或有限元前处理软件完成，步骤（8）由人工或有限元后处理软件完成。在有限元分析过程中，为达到提高计算精度、节约计算存储量、缩短计算机运算时间等目的，还应注意下列事项。

一、对称性的利用

在划分网格前要先研究一下，计算对象是否有对称变形或反对称变形存在，从而确定是取整个物体，还是取部分物体作为计算模型。如图 6.12 所示受纯弯曲的梁，结构对于 x 和 y 轴是对称的，而荷载关于 y 轴是对称的，对于 x 轴反对称。因此，应力和变形于 y 轴是对称的，关于 x 轴是反对称的。因此只需要取其中的 1/4 进行分析。但要注意两点：（1）必须在对称或反对称轴上设置相应的边界条件。在对称轴线上，垂直于对称轴的位移为零，平行于对称轴的剪力为零；在反对称轴线上，垂直于对称轴的力为零，沿反对称轴线方向的位移为零。（2）作用在对称轴线上的集中力和线分布力也应按对称性进行适当处理，例如中心受集中荷载作用的周边简支方板，取 1/4 进行有限元分析时，集中荷载也取 1/4。

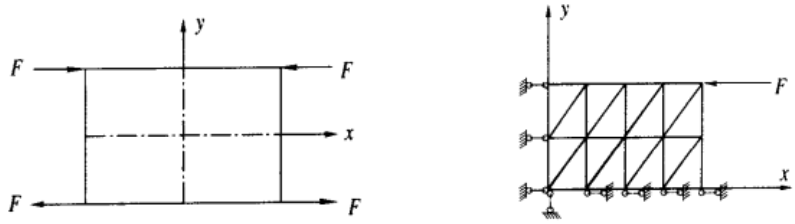


图 6.12 对称性利用

二、节点的选择和单元的划分

节点的布置和单元的划分是相互联系的。通常集中载荷的作用点、分布载荷强度的突变点、约束条件变化点都应取作节点。如果物体的厚度有突变或者物体由不同材料组成时，在布置节点时应注意，不要把厚度不同或材料不同的区域划在同一个单元里。至于节点的多少和分布的疏密，亦即单元的大小，要根据计算精度、计算机的容量、计算机计算精度、计算机的计算速度等综合考虑。网格划分越细密，有限元模型带来的误差就越小，但对计算机硬件的要求就越高。同时截断误差会由于计算量的增加而不断累加。因此，在网格划分时，应争取以较小的有限元规模获得足够的计算精度。在应力变化急剧的区域，单元划分得细一些。

平面单元的顶角不要悬殊太大，如图 6.13 所示的两种不同网格划分，虽然涉及了同样的四个节点，(a)中的两个单元的顶角相差不大，均为锐角，而(b)中两个单元均为钝角三角形，因此剖分(a)的网格质量比剖分(b)好。

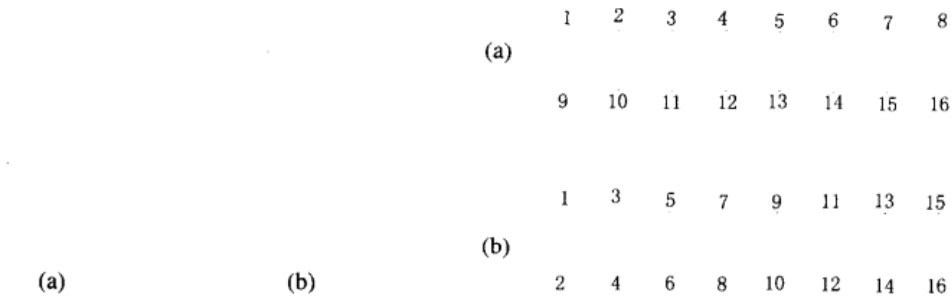


图 6.13　网格质量比较示意图

图 6.14　节点编号合理性比较

三、单元和节点的编号

前面已经指出，有限元分析过程中，方程的求解占据大量时间，如何减少方程组求解的运算量是提高有限元分析速度的关键。若总体刚度阵采用变带宽压缩存放，采用三角分解法求解线性方程组，则在节点编号时应注意使同一单元的节点号码差尽可能小，以便缩小刚度矩阵的带宽，节约计算机的存贮容量和提高运算效率。考虑图 6.14 所示(a)、(b)两种不同节点编号形式，编号方法(b)要比(a)节约许多存储空间。

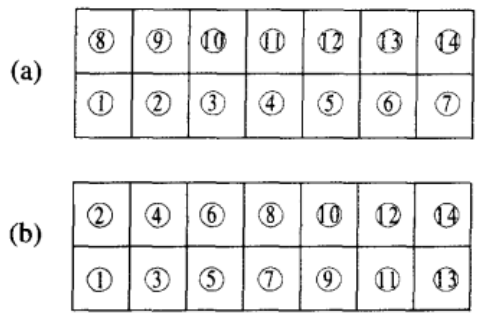


图 6.15　单元编号合理性比较

如果采用波阵法求解线性方程组，则应注意使相邻单元的编号差尽可能小，如图 6.15(b)所示的单元编号要比 6.15(a)所示的单元编号合理。

四、边界条件处理和整体刚度阵的修正

对于静力学问题，整体刚度阵本身是奇异矩阵，因此必须考虑边界约束条件，排除弹性体的刚体位移。只有在消除整体刚度矩阵的奇异性后，才能求解节点位移。在一般情况下，所考虑问题的边界往往已有一定的位移约束条件，排除了刚体运动的可能性。否则，可适当指定某些点的位移，以避免出现刚体运动。引用这些边界条件以后，待求节点未知量的数目和方程的数目便可相应地减少，然而，比较方便的做法是以某种方式引入已知的节点位移，而保持方程原有的数目不变，以避免计算机存储作大的更动。下面就介绍两种比较简便的引入已知节点位移的方法。

一种方法是把节点位移的指定值置入线性方程组中，保持方程阶数，而将 $\boldsymbol{K}$ 和 $\boldsymbol{F}$ 修正。例如，若指定有限元系统的第 i 个自由度的值为 d_i，则令 $\boldsymbol{K}$ 中的第 i 行 i 列元素 K_{ii} 为 1，而第

i 行和第 i 列的其余元素都为零。$\boldsymbol{F}$ 中的第 i 个元素则用指定位移 d_i 代入，$\boldsymbol{F}$ 中的其他各行元素都减去指定值和原来 $\boldsymbol{K}$ 中这行的相应列元素的乘积。

为了说明这一引进节点已知位移的过程，我们来考察下面只有四个方程的简单例子。

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ F_{x2} \\ F_{y2} \end{Bmatrix} \tag{6.130}$$

设这个系统中节点位移 u_1 和 u_2 被指定为

$$u_1 = \beta_1 \quad u_2 = \beta_3$$

当引用上述方法后，方程（6.130）就变成，

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & K_{22} & 0 & K_{24} \\ 0 & 0 & 1 & 0 \\ 0 & K_{42} & 0 & K_{44} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{Bmatrix} = \begin{Bmatrix} \beta_1 \\ F_{y1} - K_{21}\beta_1 - K_{23}\beta_3 \\ \beta_3 \\ F_{y2} - K_{41}\beta_1 - K_{43}\beta_3 \end{Bmatrix}$$

然后，用这组维数不变的方程来求解所有的节点位移。显然，其解答满足 $u_1 = \beta_1$、$u_2 = \beta_3$，而 v_1、v_2 仍为原方程的解答。

另一种将指定的节点位移值置入线性方程组的方法，是将 $\boldsymbol{K}$ 中与指定节点位移有关的主对角元素乘上一个大数，例如 1×10^{15}，同时将 $\boldsymbol{F}$ 的对应元素换上节点位移指定值与同一个大数的乘积。实际上，这种方法就是使得 $\boldsymbol{K}$ 中相应行的修正项远大于非修正项。

若用此方法来修正上面的例子，则方程（6.130）将成为

$$\begin{bmatrix} K_{11}\times10^{15} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33}\times10^{15} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{Bmatrix} = \begin{Bmatrix} \beta_1 K_{11}\times10^{15} \\ F_{y1} \\ \beta_3 K_{33}\times10^{15} \\ F_{y2} \end{Bmatrix}$$

为了看出此方程能给出所需的结果，我们来考察这方程组的第一个方程，

$$K_{11}\times10^{15}u_1 + K_{12}v_1 + K_{13}u_2 + K_{14}v_2 = \beta_1 K_{11}\times10^{15}$$

因为，$K_{11}\times10^{15} >> K_{ij}$，因此上式就与下式相同，

$$u_1 = \beta_1$$

上述两种方法都保持了整体刚度矩阵原有的特性，且涉及的数据量少。

参考文献

[1] 丁皓江，何福保，谢贻权，徐兴.弹性和塑性力学中的有限单元法. 北京: 机械工业出版社，1989

[2] 王焕定，吴德伦.有限单元法及计算程序. 北京: 中国工业出版社，1997

[3] O.C. Zienkiewicz. The finite element method. NewYork: Mc Graw-Hill, Inc, 1977

[4] 王勖成.有限单元法.北京:清华大学出版社，2003

第7章

改进和退化的等参单元

前一章介绍了线性有限元求解的一般过程，重点叙述了等参数单元的性质、构造以及相关的有限元列式。本章介绍作者近年来在等参单元的改进和推广应用方面的工作[1~14]。

7.1 改进的等参单元

上一章曾经提到在划分单元时，不要把厚度不同或材料不同的区域划在同一单元里，以期取得较为精确的结果。在工程应用中，这样的限制有时是非常不便的。比如，分析层合结构的低阶固有频率和固有模态时，可以将单元划分得粗放些。为此，同一单元中应允许包含有若干不同的材料区域，此时单元元素矩阵的积分必须分区进行。下面以4～8节点平面单元和8～20节点空间单元为例介绍改进的等参单元。

7.1.1 改进的平面等参单元

根据需要将四边形等参单元划分成若干区块，不失一般性，假定每个区块均为四边形，如图7.1示。同4～8节点等参单元相似，假设每个区块的形状可以用4～8个顶点来描述，将三角形区块看成四边形区块的特例。描述区块形状的顶点和单元节点不同：第一，节点的位置是用整体坐标表示的，虽然顶点也可以用整体坐标表示，但为应用方便起见，它通常定义在母单元的自然坐标系中。第二，除位移约束的节点外，节点位移是独立待求的，而顶点的位移是不独立的，它由单元节点位移插值惟一确定。

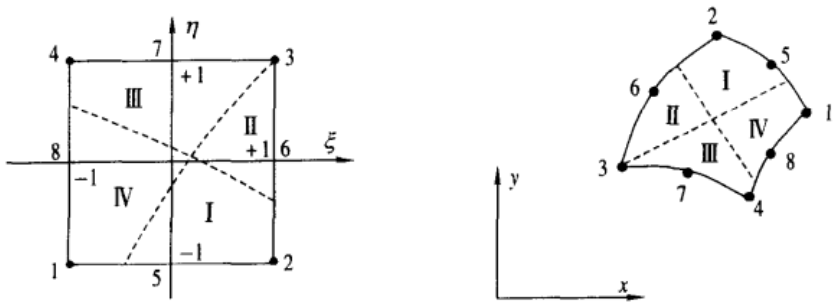


图 7.1 改进平面等参元

记第k个区块的第i个顶点的母单元坐标为$\left(\xi_i^k,\eta_i^k\right)$。为便于在各个区域内采用Gauss积分，第$k$个区块内任意点的自然坐标$(\xi,\eta)$也采用单元坐标插值相同的方法进行插值，

$$\xi = \sum_{i=1}^{n^k} N_i(\xi', \eta')\xi_i^k \tag{7.1}$$

$$\eta = \sum_{i=1}^{n^k} N_i(\xi', \eta')\eta_i^k \tag{7.2}$$

式中，n^k 为描述第 k 个区块的顶点数目，$\xi' \in [-1,1], \eta' \in [-1,1]$，$N_i$ 为 4～8 节点平面单元的形函数。式(7.1)和(7.2)在单元中的每一个区块和一边长为 2 的正方形间建立起一种一一对应的映射关系。我们称这样的正方形为母区块，母区块和母单元具有相似的性质。类似地，我们可以得到区块和母区块间坐标变换的雅可比矩阵，

$$\boldsymbol{J}' = \begin{bmatrix} \xi_{\xi'} & \eta_{\xi'} \\ \xi_{\eta'} & \eta_{\eta'} \end{bmatrix} \tag{7.3}$$

式中，

$$\begin{aligned} \xi_{\xi'} &= \frac{\partial \xi}{\partial \xi'} = \sum_{i=1}^{n^k} \frac{\partial N_i}{\partial \xi'} \xi_i^k & \xi_{\eta'} &= \frac{\partial \xi}{\partial \eta'} = \sum_{i=1}^{n^k} \frac{\partial N_i}{\partial \eta'} \xi_i^k \\ \eta_{\xi'} &= \frac{\partial \eta}{\partial \xi'} = \sum_{i=1}^{n^k} \frac{\partial N_i}{\partial \xi'} \eta_i^k & \eta_{\eta'} &= \frac{\partial \eta}{\partial \eta'} = \sum_{i=1}^{n^k} \frac{\partial N_i}{\partial \eta'} \eta_i^k \end{aligned} \tag{7.4}$$

根据积分的线性性质，单元的刚度矩阵(6.64)式可改写为

$$\boldsymbol{K}^e = \sum_{k=1}^{nm} \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}^T \boldsymbol{D}^k \boldsymbol{B} |\boldsymbol{J}| |\boldsymbol{J}'| \mathrm{d}\xi' \mathrm{d}\eta' \tag{7.5}$$

式中，nm 为单元中区块数目，$|\boldsymbol{J}'|$ 为区块和母区块间坐标变换雅可比行列式的值，$\boldsymbol{D}^k$ 为第 k 个区块的弹性系数矩阵。如果某个区块的 $\boldsymbol{D}^k = \boldsymbol{0}$，则表示该区块为空，即为“空气”区块。

从形式上看，式（7.1）和（7.2）仅仅是一种数学变换，但这种变化为有限单元法的应用带来很大的方便。最明显的优点就是（7.5）式允许同一单元由不同的材料构成，特别适合类似层合结构等问题的总体特性分析。它的另外一个非常重要的优点，就是把有限元网格（单元和节点）和问题的物理区域独立开来。网格的剖分不再严格依赖于物理区域，我们可以把节点及其坐标定义在结构的外部，只需将单元位于结构外的部分区域定义为“空气”区块就可以了。根据习惯，一般有 $\xi_i^k \in [-1,1], \eta_i^k \in [-1,1]$，但这个条件在（7.4）式中并不是必要的，当顶点的自然坐标超越这样的范围时，表示该顶点在单元之外，区块的全部或部分在单元所覆盖的区域之外。也就是说，采用改进的等参数单元，允许部分结构位于有限元网格之外。

改进的等参数单元使得网格的自动剖分成为一件非常容易实现的工作。以图 7.2 所示结构为例，图 7.2（a）为一待离散的物理区域，有限元离散时先确定一覆盖该物理区域的长方形，如图 7.2(b)所示，将该矩形区域进行网格剖分，得图 7.2(c)所示网格，最后分析有限单元和物理区域的交集以确定单元内区块。如果单元和物理区域的交集为空，则该单元可以删除。

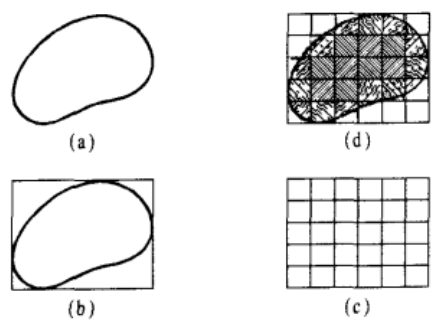


图 7.2 改进等参元的网格剖分

和单元刚度矩阵相似，单元的一致质量阵为

$$\boldsymbol{M}^e=\sum_{k=1}^{nm}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{N}^T\boldsymbol{N}\rho^k|\boldsymbol{J}||\boldsymbol{J}'|\mathrm{d}\xi'\mathrm{d}\eta' \tag{7.6}$$

单元的集中质量阵为

$$\boldsymbol{M}^e=\mathrm{diag}(m_1,m_1,m_2,m_2,\cdots,m_n,m_n) \tag{7.7}$$

$$m_i=\sum_{k=1}^{nm}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{N}_i\rho^k|\boldsymbol{J}||\boldsymbol{J}'|\mathrm{d}\xi'\mathrm{d}\eta' \tag{7.8}$$

单元的体积力等效节点力为

$$\boldsymbol{F}_f^e=\sum_{k=1}^{nm}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{N}^T\boldsymbol{f}^k|\boldsymbol{J}||\boldsymbol{J}'|\mathrm{d}\xi'\mathrm{d}\eta' \tag{7.9}$$

但要注意的是，作用在结构表面的外力不一定作用在单元的表面，因此计算单元“表面力”等效节点力时不一定有$\xi=\pm1$或$\eta=\pm1$。但在力作用的“边”上，ξ和η应该满足某种关系，不妨用函数表示为

$$b(\xi,\eta)=0 \tag{7.10}$$

对(7.10)式全微分，

$$\frac{\partial b}{\partial\xi}\mathrm{d}\xi+\frac{\partial b}{\partial\eta}\mathrm{d}\eta=0 \tag{7.11}$$

如果$\dfrac{\partial b}{\partial\eta}\neq0$，则有

$$\eta_\xi=\frac{\partial\eta}{\partial\xi}=-\frac{\partial b}{\partial\xi}\Big/\frac{\partial b}{\partial\eta} \tag{7.12}$$

此时，

$$\mathrm{d}\Gamma=\sqrt{(x_\xi+x_\eta\eta_\xi)^2+(y_\xi+y_\eta\eta_\xi)^2}\mathrm{d}\xi \tag{7.13}$$

否则，

$$\xi_\eta=\frac{\partial\xi}{\partial\eta}=-\frac{\partial b}{\partial\eta}\Big/\frac{\partial b}{\partial\xi} \tag{7.14}$$

则有

$$\mathrm{d}\Gamma=\sqrt{(x_\xi\xi_\eta+x_\eta)^2+(y_\xi\xi_\eta+y_\eta)^2}\mathrm{d}\eta \tag{7.15}$$

利用（7.13）和 （7.15）式容易求得表面力等效节点力。

7.1.2 改进的空间等参单元

将改进平面等参元推广到 8～20 节点空间等参元，容易得到改进的空间等参单元。同样，根据需要将边长为 2 的立方体母单元划分成若干区块。不失一般性，假定每个区块都是六面体，可以用 8～20 个顶点描述其几何形状。记第 k 个区块的第 i 个顶点的母单元坐标为$\left(\xi_i^k,\eta_i^k,\zeta_i^k\right)$。区块内任意点的自然坐标可用类似单元坐标插值的方法表示为

$$\xi=\sum_{i=1}^{n^k}N_i\left(\xi',\eta',\zeta'\right)\xi_i^k$$

$$\eta=\sum_{i=1}^{n^k}N_i(\xi',\eta',\zeta')\eta_i^k \tag{7.16}$$

$$\zeta=\sum_{i=1}^{n^k}N_i(\xi',\eta',\zeta')\zeta_i^k$$

式中，n^k 为描述第 k 个区块需要的顶点数，取值 8～20。$\xi'\in[-1,1],\eta'\in[-1,1],\zeta'\in[-1,1]$ 为母区块的自然坐标。区块和母区块间坐标变换的雅可比矩阵为

$$\boldsymbol{J}'=\begin{bmatrix}\xi_{\xi'} & \eta_{\xi'} & \zeta_{\xi'}\\ \xi_{\eta'} & \eta_{\eta'} & \zeta_{\eta'}\\ \xi_{\zeta'} & \eta_{\zeta'} & \zeta_{\zeta'}\end{bmatrix} \tag{7.17}$$

式中，

$$\begin{aligned}
&\xi_{\xi'}=\frac{\partial\xi}{\partial\xi'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\xi'}\xi_i^k \quad \xi_{\eta'}=\frac{\partial\xi}{\partial\eta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\eta'}\xi_i^k \quad \xi_{\zeta'}=\frac{\partial\xi}{\partial\zeta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\zeta'}\xi_i^k\\
&\eta_{\xi'}=\frac{\partial\eta}{\partial\xi'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\xi'}\eta_i^k \quad \eta_{\eta'}=\frac{\partial\eta}{\partial\eta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\eta'}\eta_i^k \quad \eta_{\zeta'}=\frac{\partial}{\partial\zeta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\zeta'}\eta_i^k\\
&\zeta_{\xi'}=\frac{\partial\zeta}{\partial\xi'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\xi'}\zeta_i^k \quad \zeta_{\eta'}=\frac{\partial\zeta}{\partial\eta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\eta'}\zeta_i^k \quad \zeta_{\zeta'}=\frac{\partial\zeta}{\partial\zeta'}=\sum_{i=1}^{n^k}\frac{\partial N_i}{\partial\zeta'}\zeta_i^k
\end{aligned} \tag{7.18}$$

单元的刚度矩阵为

$$\boldsymbol{K}_e=\sum_{k=1}^{nm}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{B}^T\boldsymbol{D}^k\boldsymbol{B}|\boldsymbol{J}||\boldsymbol{J}'|\mathrm{d}\xi'\mathrm{d}\eta'\mathrm{d}\zeta' \tag{7.19}$$

式中，nm 为区块数，$|\boldsymbol{J}'|$ 为区块与母区块间坐标变换的雅可比行列式的值，$\boldsymbol{D}^k$ 为第 k 个区块的弹性系数矩阵。

改进等参元的单元质量阵和体积力等效节点力等也可以类似平面问题计算，这里不再重复。同样，改进的空间等参单元将大大降低空间问题网格自动剖分的困难，具有十分重要的实用价值。

7.2 改进的等参元在自由面渗流分析中应用[4]

改进等参元由于其灵活性，除在网格剖分方面具有很多优点外，还在大量实际工程问题中得到应用。这里介绍其在具有自由面的渗流问题中的应用。

边坡渗流、坝体渗流等一般都是涉及有自由面的渗流问题，其作为渗流域边界的自由面是待求的，即渗流范围未知，因而这类问题的分析是非线性的，需要反复迭代。目前求解这类问题有两大类方法。

第一大类是移动网格法。这类方法把自由面当作可移动边界处理，在迭代过程中不断改变自由面的位置，使网格发生变化，直到自由面位置稳定为止[15][16]。采用这类方法虽然取得了不少成功的经验，但也暴露出方法本身的缺陷[17]：a). 当初始自由面和最终自由面相差较大时，网格过分变形导致网格畸形；b). 渗流域内有结构物时网格移动会改变结构边界；c). 自由面附近有不均质介质时，网格移动时会破坏介质区分边界，使计算结果失真；d). 求解包含有自由面渗流域的应力分布时，求解范围应包括自由面以上区域，移动网格法不能采用同一

网格分析渗流场和应力场，增加了有自由面渗流问题应力分析的工作量。由于存在一些严重的缺点，移动网格法已逐步被淘汰。

第二大类是固定网格法。为克服移动网格法存在的问题，国内外学者致力于寻找有自由面渗流分析的新方法，其研究核心就是计算中不改变网格，即固定网格法。自 1973 年 Neuman 提出用不变网格分析有自由面渗流的 Galerkin 法以来，出现了多种固定网格法。固定网格法又可以分成两种，一种是调整单元渗透矩阵法，如 Bathe(1979)的单元渗透矩阵调整法[18]，李春华(1986)的改进单元渗透矩阵调整法[19]，王贤能等(1997)的高斯点法[20]。Bathe 法在迭代过程中把跨越自由面和自由面以上的单元的渗透系数一律按实际渗透系数的千分之一计算，与实际不吻合。李春华则在迭代中根据自由面和单元的交截情况计算单元的渗透系数。王贤能则根据高斯点的位置确定高斯点对应的渗透系数，计算单个高斯点对渗透矩阵的贡献，对所有高斯点求和得到单元的渗透矩阵。另一种是调整流量法，如 C.S.Desai(1976)提出的剩余流量法[21]，张有天(1988)的初流量法[22]等。Desai 法通过计算自由面单元内过自由表面的流量，来修改各节点的势，直到过自由面的流量小于某一允许值为止。Desai 法的缺点是每一迭代步需要计算出自由面的具体位置、自由面的面积以及法向流量，工作量大，而且这一方法的全部调整过程均依赖于第一次有限元分析结果，计算精度差。张有天的初流量法对 Desai 法作了许多重大改进，并从理论上给出了完整的描述。

另外，吴梦喜等(1994)提出了虚单元法[17]，这一方法的本质是在固定网格基础上的移动网格法，计算过程中仍要不断地移动网格。梁业国等(1997)的子单元法[23]则将跨越自由面的单元细划，提出子单元的概念。这一方法在具体实现时需要增减单元和节点。

在上述研究的基础上，我们利用改进的等参单元提出了有自由面渗流分析的虚节点法。虚节点法根据上一步分析得到的自由面位置，把分析区域分成三部分。对于自由面以下的单元，取积分区域为整单元；自由面以上的单元对渗透矩阵没有贡献，无需积分；对于跨越自由面的单元，自由面把单元划分成两个部分，对渗透矩阵有贡献的区域在自由面以下。自由面以下部分可以根据其形状划分成若干个积分区块，分别进行积分。虚节点法是一种标准的固定网格法，对网格不作任何改动，并能精确地描述跨越自由面的单元的渗透矩阵。

采用上述原理，作者在有限元分析软件 VFEAP 中开发了有自由面渗流分析单元，并利用该单元分析了两个算例，均得到了很好的效果。

例 7-1　如图 7.3 所示，一各向同性正方形断面均质土坡，上游水头为 10m，下游水头为 2m。将整个区域划分为 100 个单元。经过五次迭代分析后，最终自由面的位置如图 7.3 所示。结果与解析解以及梁业国等由子单元法得到的结果完全一致。

例 7-2　如图 7.4 所示一梯形土坝，上游水头为 60m，下游无水，介质为均质各向同性。整个计算区域如图 7.4 所示划分网格，最终自由面位置与解析解完全一致。

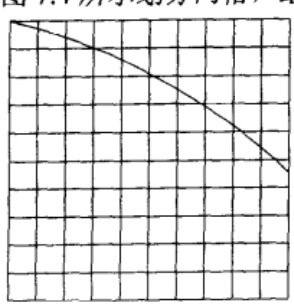


图 7.3 正方形土坡自由面

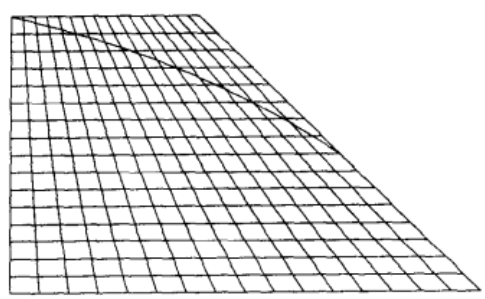


图 7.4　梯形土坝自由面

7.3 退化的等参单元

本节在 8～20 节点空间等参单元中适当地引入梁板壳的基本假定，构造了退化梁板壳单元系列，并应用于梁板壳等结构分析。

7.3.1 退化等参单元的构造思路

基于经典板壳理论构造位移协调的板壳单元存在一定的困难，板壳单元的构造一直是有限元领域中研究得最多的问题，出现了各种各样的板壳单元，如 3 节点 9 自由度部分协调元，3 节点 18 自由度或 6 节点 21 自由度过协调单元，4 节点 12（或 24）自由度矩形过协调单元；用二次分片插值法的 3 节点 9 自由度三角形协调元，4 节点 12 自由度四边形协调单元；用杂交法建立的各种杂交单元；用条件极值法建立的协调单元；用分项插值法建立的协调单元等等。对于三广义位移的板壳理论，也有多种单元，最常用的是 8 节点 24 自由度的等参元。有大量的文献和著作阐述了板壳单元的构造原理和工程应用。

虽然基于板壳理论得到的板壳单元种类繁多，且仍层出不穷，然而由于对简洁明了的等参单元的偏爱，人们一直希望把板壳结构当作一种特殊的块体结构，采用三维实体单元或类似于三维实体单元的等参单元进行分析，一些著名学者进行过这方面的深入研究。R.D.Wood 直接采用三维块体单元对板壳结构进行了分析，他在研究中注意到了扁平单元会使刚度矩阵变得病态，因此提出相对位移的概念。遗憾的是，数值结果表明 Wood 的试验是失败的，他得到了错误的结果。因此，O.C.Ziekiewicz 指出，采用这种单元，“当泊松比不为零时，结果由于 $(1-\nu^2)$ 而收敛于一个错误的解答”。徐兴等对这类单元进行了深入的研究，发现 Wood 的研究得到一个错误解答的原因是他没有适当地引入板壳理论中相应的假定。

众所周知，经典的板壳单元都是根据板壳理论构造出来的，而经典的板壳理论则是一般的三维弹性理论根据板壳结构特殊的几何形状和变形特点引入一定的简化假定后得到的，因此可以认为板壳理论是一种特定条件下简化的三维弹性理论。注意，引入板壳假定后的板壳理论都是基于板壳的中面的。经典的板壳单元正是基于板壳理论构造的，它们也都是基于中面的，只需要在板壳结构的中面剖分有限元网格。由此可见，经典的板壳单元走的是先理论后单元的道路，即先引入板壳假定，从三维弹性理论得到板壳理论，然后构造单元。实际上我们也可以走另外一条路径构造板壳单元，即先单元后理论。具体地说就是，从三维弹性理论出发，构造三维实体单元，再在三维实体单元的基础上引入板壳的基本假定，使之适用于板壳结构分析。也就是说,板壳单元可以看成是一种特定条件下的三维实体单元,如图 7.5 所示。

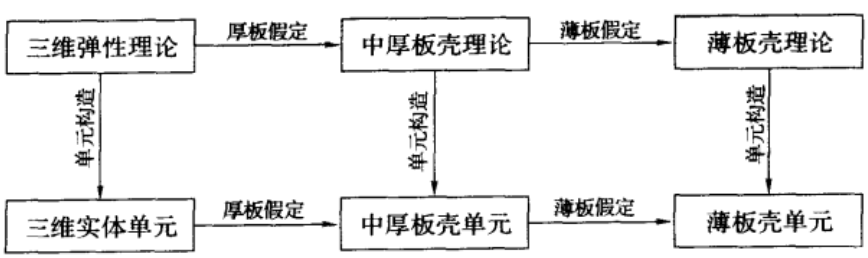


图 7.5 板壳理论与板壳单元

7.3.2　相对位移板单元[1]

如图 7.6 所示的 16 节点板单元，假定坐标轴 z 沿板的法线方向。每个节点有 u、v、w 三个自由度，共 48 个自由度。单元的节点位移矢量为

$$\boldsymbol{d}^e=\{u_1,v_1,w_1,\cdots,u_5,v_5,w_5,\cdots w_{16}\}^T \tag{7.20}$$

采用与三维实体单元相同的单元坐标和位移插值形函数。

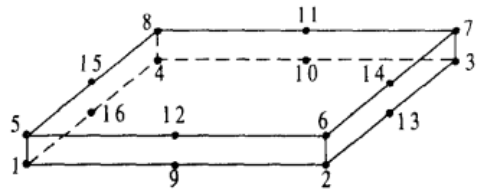


图7.6　16节点板单元

一、相对位移

考虑到扁平单元会使刚度矩阵病态，采用 R.D.Wood 的建议，用相对位移的办法克服。引入相对位移，将上表面所有节点的位移表示为相应的下表面节点位移和相对位移之和，以节点 5 为例，

$$\begin{aligned}u_5&=u_1+\Delta u_5\\v_5&=v_1+\Delta v_5\\w_5&=w_1+\Delta w_5\end{aligned} \tag{7.21}$$

将 $\Delta u_5,\Delta v_5,\Delta w_5$ 作为节点 5 的独立的未知位移量。于是单元节点位移矢量改写为

$$\boldsymbol{d}^e=\{u_1,v_1,w_1,\cdots,\Delta u_5,\Delta v_5,\Delta w_5,\cdots,w_{16}\}^T \tag{7.22}$$

二、板壳假定的引入

（1）中厚板假定

根据板理论，应力满足 $\sigma_z \ll \sigma_x$，$\sigma_z \ll \sigma_y$，因此 σ_z 产生的变形可以忽略不计，即在物理方程中可认为 $\sigma_z=0$。引入这一假定后，弹性应力应变关系简化为

$$\begin{Bmatrix}\sigma_x\\\sigma_y\\\tau_{yz}\\\tau_{zx}\\\tau_{xy}\end{Bmatrix}=\begin{bmatrix}d_1&d_2&0&0&0\\&d_1&0&0&0\\&&d_3&0&0\\&对称&&d_3&0\\&&&&d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\\varepsilon_y\\\gamma_{yz}\\\gamma_{zx}\\\gamma_{xy}\end{Bmatrix} \tag{7.23}$$

$$\varepsilon_z=-\nu(\varepsilon_x+\varepsilon_y) \tag{7.24}$$

式中，

$$d_1=\frac{E}{1-\nu^2}\qquad d_2=\nu d_1\qquad d_3=G=\frac{E}{2(1+\nu)} \tag{7.25}$$

从式(7.24)可以看出，ε_z 为不独立的应变分量，将（7.24）式沿厚度方向积分，得相对挠度，

$$\Delta w=\int_{-h/2}^{h/2}\varepsilon_z\mathrm{d}z=-\nu\int_{-h/2}^{h/2}(\varepsilon_x+\varepsilon_y)\mathrm{d}z \tag{7.26}$$

式中，h 为板厚。显然，Δw 也是不独立的量，所以，即使去掉所有刚体位移，由（7.23）式得到的刚度矩阵仍是奇异的，由此得出的解也可能是错误的。为克服 Δw 不独立导致的奇异性，在有限元求解过程中，约束 Δw 对应的自由度，即假定

$$\Delta w=0 \tag{7.27}$$

约束自由度 Δw 后，单元自由度数总数从 48 降至 40，与 8 节点 40 自由度厚壳单元相同，而且

有限元列式更加简单统一。

如果引入中面不伸长的假定，

$$u_{上}+u_{下}=0 \tag{7.28}$$

$$v_{上}+v_{下}=0 \tag{7.29}$$

式中，下标“上”和“下”分别代表上表面和下表面对应的节点。式(7.28)和(7.29)又将约束8×2=16自由度，单元变成16节点24自由度厚板单元，与8节点24自由度厚板元相同。

（2）Kirchhoff板假定

根据Kirchhoff板理论假定，$\gamma_{xz}=\gamma_{yz}=0$，即薄板横向的剪切刚度无限大，为此在(7.23)式中对相应的刚度系数进行修正，即乘以一个大数λ(一般取$\lambda=1000$)。此时应力应变关系修正为

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{yz}\\ \tau_{zx}\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ & d_1 & & & & \\ & & & \lambda d_3 & & \\ & 对称 & & & \lambda d_3 & \\ & & & & & d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{yz}\\ \gamma_{zx}\\ \gamma_{xy}\end{Bmatrix} \tag{7.30}$$

矩阵中未填元素为零，下同。其他约束处理同中厚板。

（3）薄膜假定

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\end{Bmatrix}=\begin{bmatrix}d_1 & d_2\\ d_2 & d_1\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\end{Bmatrix} \tag{7.31}$$

同时引入约束，

$$\begin{aligned}\Delta u&=u_{上}-u_{下}=0\\ \Delta v&=v_{上}-v_{下}=0\\ \Delta w&=w_{上}-w_{下}=0\end{aligned} \tag{7.32}$$

可得到16节点24自由度膜单元。

三、单元刚度矩阵和质量阵

上述分析表明，适当地修正应力应变关系和约束一部分相对位移或中面位移可引入Mindlin板，Kirchhoff板和膜理论的基本假定。将应力应变关系式(7.23)、(7.30)和(7.31)扩阶可得与三维弹性问题相似的应力应变关系。

Mindlin板

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{yz}\\ \tau_{zx}\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ d_2 & d_1 & & & & \\ & & 0 & & & \\ & & & d_3 & & \\ & & & & d_3 & \\ & & & & & d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{yz}\\ \gamma_{zx}\\ \gamma_{xy}\end{Bmatrix} \tag{7.33}$$

Kirchhoff板

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{yz}\\ \tau_{zx}\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ d_2 & d_1 & & & & \\ & & 0 & & & \\ & & & \lambda d_3 & & \\ & & & & \lambda d_3 & \\ & & & & & d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{yz}\\ \gamma_{zx}\\ \gamma_{xy}\end{Bmatrix} \tag{7.34}$$

薄膜

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{yz}\\ \tau_{zx}\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ d_2 & d_1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{yz}\\ \gamma_{zx}\\ \gamma_{xy}\end{Bmatrix} \tag{7.35}$$

没有引入相对位移时，将(6.100)式中的 $\boldsymbol{D}$ 改成扩阶后的弹性系数矩阵，就得到相应单元的刚度矩阵。单元一致质量阵和集中质量阵仍分别为 (6.102) 式和(6.105)式。

四、相对位移的引入

下面以单元刚度矩阵为例，说明如何引入相对位移。根据相对位移的定义，可以得到节点自由度的变换关系，

$$\hat{\boldsymbol{d}}^e=\hat{\boldsymbol{T}}\,\boldsymbol{d}^e \tag{7.36}$$

式中，$\hat{\boldsymbol{T}}$ 为 48×48 的矩阵

$$\boldsymbol{d}^e=\{\boldsymbol{u}_1^T,\boldsymbol{u}_2^T,\cdots,\boldsymbol{u}_{16}^T\}^T$$

$$\hat{\boldsymbol{d}}^e=\{\boldsymbol{u}_1^T,\cdots,\boldsymbol{u}_4^T,\Delta\boldsymbol{u}_5^T,\cdots,\Delta\boldsymbol{u}_8^T,\boldsymbol{u}_9^T,\boldsymbol{u}_{10}^T,\Delta\boldsymbol{u}_{11}^T,\Delta\boldsymbol{u}_{12}^T,\boldsymbol{u}_{13}^T,\Delta\boldsymbol{u}_{14}^T,\Delta\boldsymbol{u}_{15}^T,\boldsymbol{u}_{16}^T\}^T$$

于是引入相对位移后的刚度矩阵和等效节点力分别为

$$\hat{\boldsymbol{K}}^e=\hat{\boldsymbol{T}}^T\boldsymbol{K}^e\hat{\boldsymbol{T}} \tag{7.37}$$

$$\hat{\boldsymbol{F}}^e=\hat{\boldsymbol{T}}^T\boldsymbol{F}^e \tag{7.38}$$

注意到，如果将 $\hat{\boldsymbol{T}}$ 分成 16×16 个 3×3 的子矩阵，其中绝大部分子矩阵为 $\boldsymbol{0}$，少量非 $\boldsymbol{0}$ 子矩阵或为单位矩阵，或为负的单位矩阵。因此在程序实现时并不需要进行大量的矩阵乘法运算，只要进行行列间的加减运算就可以了。

组集后求解之前要引入相对位移约束条件，以免得到错误的结果。在单元正交曲线坐标系下，在应力应变关系中引入壳的假定，可将相对位移板单元推广到壳体结构分析中，构造出 16 节点 40 自由度的壳单元。

7.3.3　实体退化单元系列

相对位移的引入有两个优点：一是克服扁平单元导致的刚度矩阵病态；二是方便地引入相对位移约束，大大地减少了线性方程组求解时间。随着计算机软硬件技术的快速发展，计算精度和计算速度大大提高。原来必须通过引入相对位移来克服的刚度矩阵病态已不再是一个问题。直接采用节点的位移而不是相对位移，可以使得梁板壳单元和三维空间等参单元不

仅在形式上，而且在计算过程上完全相同，与三维空间单元的连接也更加自然。这里我们仍然通过应力应变关系来引入梁板壳的基本假定和相对位移约束，但不再引入相对位移的概念。

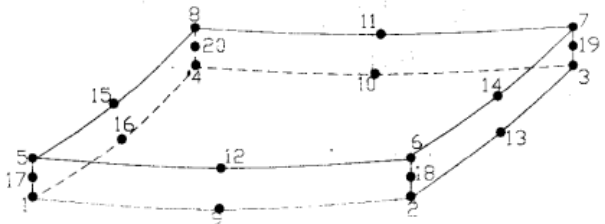


图 7.7 实体退化板壳元

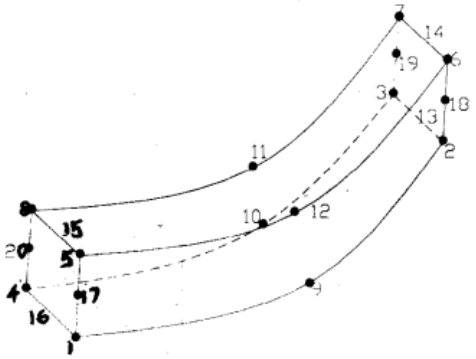


图 7.8 实体退化梁元

对于 8～20 节点等参单元，单元坐标和位移插值采用(6.88)和(6.89)式。对于板壳问题，规定节点 1、2、3、4 在板的下（或上）表面（见图 7.7），对于梁结构，规定 ξ 方向沿梁的轴线方向（见图 7.8）。为描述方便起见，在单元内任意点按以下方法建立正交曲线坐标系 $o'x'y'z'$。

(1) 计算矢量 $\boldsymbol{V}_1=\{\bar{l}_{x'},\bar{m}_{x'},\bar{n}_{x'}\}^T$ 和 $\boldsymbol{V}_2=\{\bar{l}_{y'},\bar{m}_{y'},\bar{n}_{y'}\}^T$，其中，

$$\bar{l}_{x'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\xi}x_i \quad \bar{m}_{x'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\xi}y_i \quad \bar{n}_{x'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\xi}z_i \tag{7.39}$$

$$\bar{l}_{y'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\eta}x_i \quad \bar{m}_{y'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\eta}y_i \quad \bar{n}_{y'}=\sum_{i=1}^{n}\frac{\partial N_i(\xi,\eta,0)}{\partial\eta}z_i \tag{7.40}$$

对于梁单元，还要令上式中的 $\eta=0$。

(2) x' 坐标线的单位切线方向矢量 $\boldsymbol{i}_{x'}=\{l_{x'},m_{x'},n_{x'}\}^T$，其中，

$$l_{x'}=\frac{\bar{l}_{x'}}{L_{x'}} \qquad m_{x'}=\frac{\bar{m}_{x'}}{L_{x'}} \qquad n_{x'}=\frac{\bar{n}_{x'}}{L_{x'}} \tag{7.41}$$

其中，

$$L_{x'}=\sqrt{\bar{l}_{x'}^2+\bar{m}_{x'}^2+\bar{n}_{x'}^2}$$

(3) z' 坐标线的单位切线方向矢量 $\boldsymbol{i}_{z'}=\{l_{z'},m_{z'},n_{z'}\}^T$，它由矢量 $\boldsymbol{V}_1\times\boldsymbol{V}_2$ 归一化确定。

(4) y' 坐标线的单位切线方向矢量 $\boldsymbol{i}_{y'}=\{l_{y'},m_{y'},n_{y'}\}^T$，它由 $x'y'z'$ 为右手坐标系确定，

$$\boldsymbol{i}_{y'}=\boldsymbol{i}_{z'}\times\boldsymbol{i}_{x'} \tag{7.42}$$

显然，对于梁单元，x' 为梁的轴线方向；对于板壳单元 z' 为板壳单元的法线方向。下面我们介绍梁板壳假定的引入。

一、中厚板壳单元

各向同性线性弹性材料的应力应变关系可写成，

$$\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix}=\frac{1}{E}\begin{bmatrix}1 & -\nu & -\nu & & & \\ -\nu & 1 & -\nu & & & \\ -\nu & -\nu & 1 & & & \\ & & & 2(1+\nu) & & \\ & & & & 2(1+\nu) & \\ & & & & & 2(1+\nu)\end{bmatrix}\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix} \tag{7.43}$$

由于应力满足 $\sigma_{z'} << \sigma_{x'}(\sigma_{y'})$，故 $\sigma_{z'}$ 产生的变形可以忽略不计。因此可以将上式右端矩阵中的第一行第三列、第二行第三列元素设定为零。于是，弹性应力应变关系简化为

$$\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix}=\frac{1}{E}\begin{bmatrix}1 & -\nu & 0 & & & \\ -\nu & 1 & 0 & & & \\ -\nu & -\nu & 1 & & & \\ & & & 2(1+\nu) & & \\ & & & & 2(1+\nu) & \\ & & & & & 2(1+\nu)\end{bmatrix}\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix} \tag{7.44}$$

注意到，上式中的柔度系数矩阵是一个非对称矩阵。为使之对称化，必须令第三行的第一、第二列元素为零。即

$$\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix}=\frac{1}{E}\begin{bmatrix}1 & -\nu & 0 & & & \\ -\nu & 1 & 0 & & & \\ 0 & 0 & 1 & & & \\ & & & 2(1+\nu) & & \\ & & & & 2(1+\nu) & \\ & & & & & 2(1+\nu)\end{bmatrix}\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix} \tag{7.45}$$

如此按(7.45)式计算得到的应变 $\varepsilon_{z'}$ 是不准确的。由于 $\sigma_{z'}$ 很小，$\varepsilon_{z'}$ 的计算值比真实值要小得多，相当于假定 $\varepsilon_{z'}=0$ 或 $\Delta w'=0$（即板壳的相对挠度等于零）。利用这样的本构关系可以期望得到与相对位移板壳元相一致的结果。实际上，在板壳结构分析中，由于 $\sigma_{z'}$ 很小，$\delta\varepsilon_{z'}\sigma_{z'}$ 对内力所做的虚功影响很小，因此 $\varepsilon_{z'}$ 的真实大小并不重要。对(7.45)式求逆，得 d_1，d_2，d_3 见式(7.25)。

$$\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ d_2 & d_1 & & & & \\ & & E & & & \\ & & & d_3 & & \\ & & & & d_3 & \\ & & & & & d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix} \tag{7.46}$$

二、Kirchhoff 薄板壳单元

根据 Kirchhoff 板理论假定，$\gamma_{xz'}=\gamma_{y'z'}=0$，即薄板横向的剪切刚度无限大，为此修正(7.46)式中相应的刚度系数，即乘以一个大数 λ(一般取 $\lambda=1000$)。此时应力应变关系修正为

$$\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix}=\begin{bmatrix}d_1 & d_2 & & & & \\ d_2 & d_1 & & & & \\ & & E & & & \\ & & & \lambda d_3 & & \\ & & & & \lambda d_3 & \\ & & & & & d_3\end{bmatrix}\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix} \tag{7.47}$$

实际算例表明，（7.46）式适用于厚薄板单元，一般情况下不用(7.47)式，除非用薄板理论分析厚板问题。

三、梁

对于梁，应力满足如下关系，

$$\sigma_{z'}(\sigma_{y'})<<\sigma_{x'}$$

因此，$\sigma_{y'}$ 和 $\sigma_{z'}$ 产生的变形都可以忽略不计。同板壳单元相似，我们可以得到梁单元的应力应变关系为

$$\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix}=\begin{bmatrix}E & & & & & \\ & E & & & & \\ & & E & & & \\ & & & G & & \\ & & & & G & \\ & & & & & G\end{bmatrix}\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix} \tag{7.48}$$

如果不考虑梁的横向剪切变形（细长梁），则有

$$\begin{Bmatrix}\sigma_{x'}\\ \sigma_{y'}\\ \sigma_{z'}\\ \tau_{y'z'}\\ \tau_{z'x'}\\ \tau_{x'y'}\end{Bmatrix}=\begin{bmatrix}E & & & & & \\ & E & & & & \\ & & E & & & \\ & & & \lambda G & & \\ & & & & \lambda G & \\ & & & & & G\end{bmatrix}\begin{Bmatrix}\varepsilon_{x'}\\ \varepsilon_{y'}\\ \varepsilon_{z'}\\ \gamma_{y'z'}\\ \gamma_{z'x'}\\ \gamma_{x'y'}\end{Bmatrix} \tag{7.49}$$

四、曲线杆单元

由于杆单元只考虑轴向变形，将（7.48）式中的剪切模量赋值为零，就得到杆单元在局部坐标系下的应力应变关系。计算单元刚度和质量矩阵时，只要在一个方向进行数值积分就可以了。

将式(7.46)～(7.49)写成如下统一形式，

$$\boldsymbol{\sigma}'=\boldsymbol{D}'\boldsymbol{\varepsilon}' \tag{7.50}$$

根据弹性力学的知识有

$$\boldsymbol{\sigma}'=\boldsymbol{T}_\sigma\boldsymbol{\sigma}\qquad\boldsymbol{\varepsilon}'=\boldsymbol{T}_\varepsilon\boldsymbol{\varepsilon} \tag{7.51}$$

式中，

$$\boldsymbol{T}_\sigma=\begin{bmatrix} l_{x'}l_{x'} & m_{x'}m_{x'} & n_{x'}n_{x'} & 2m_{x'}n_{x'} & 2n_{x'}l_{x'} & 2l_{x'}m_{x'} \\ l_{y'}l_{y'} & m_{y'}m_{y'} & n_{y'}n_{y'} & 2m_{y'}n_{y'} & 2n_{y'}l_{y'} & 2l_{y'}m_{y'} \\ l_{z'}l_{z'} & m_{z'}m_{z'} & n_{z'}n_{z'} & 2m_{z'}n_{z'} & 2n_{z'}l_{z'} & 2l_{z'}m_{z'} \\ l_{y'}l_{z'} & m_{y'}m_{z'} & n_{y'}n_{z'} & m_{y'}n_{z'}+m_{z'}n_{y'} & n_{y'}l_{z'}+n_{z'}l_{y'} & l_{y'}m_{z'}+l_{z'}m_{y'} \\ l_{x'}l_{z'} & m_{x'}m_{z'} & n_{z'}n_{x'} & m_{x'}n_{z'}+m_{z'}n_{x'} & n_{x'}l_{z'}+n_{z'}l_{x'} & l_{x'}m_{z'}+l_{z'}m_{x'} \\ l_{x'}l_{y'} & m_{x'}m_{y'} & n_{x'}n_{y'} & m_{x'}n_{y'}+n_{x'}m_{y'} & n_{x'}l_{y'}+n_{y'}l_{x'} & l_{x'}m_{y'}+l_{y'}m_{x'} \end{bmatrix} \tag{7.52}$$

$$\boldsymbol{T}_\varepsilon=\begin{bmatrix} l_{x'}l_{x'} & m_{x'}m_{x'} & n_{x'}n_{x'} & m_{x'}n_{x'} & n_{x'}l_{x'} & l_{x'}m_{x'} \\ l_{y'}l_{y'} & m_{y'}m_{y'} & n_{y'}n_{y'} & m_{y'}n_{y'} & n_{y'}l_{y'} & l_{y'}m_{y'} \\ l_{z'}l_{z'} & m_{z'}m_{z'} & n_{z'}n_{z'} & m_{z'}n_{z'} & n_{z'}l_{z'} & l_{z'}m_{z'} \\ 2l_{y'}l_{z'} & 2m_{y'}m_{z'} & 2n_{y'}n_{z'} & m_{y'}n_{z'}+m_{z'}n_{y'} & n_{y'}l_{z'}+n_{z'}l_{y'} & l_{y'}m_{z'}+l_{z'}m_{y'} \\ 2l_{x'}l_{z'} & 2m_{x'}m_{z'} & 2n_{z'}n_{x'} & m_{x'}n_{z'}+m_{z'}n_{x'} & n_{x'}l_{z'}+n_{z'}l_{x'} & l_{x'}m_{z'}+l_{z'}m_{x'} \\ 2l_{x'}l_{y'} & 2m_{x'}m_{y'} & 2n_{x'}n_{y'} & m_{x'}n_{y'}+n_{x'}m_{y'} & n_{x'}l_{y'}+n_{y'}l_{x'} & l_{x'}m_{y'}+l_{y'}m_{x'} \end{bmatrix} \tag{7.53}$$

将(7.51)式代入(7.50)式得

$$\boldsymbol{\sigma}=\boldsymbol{T}_\sigma^{-1}\boldsymbol{D}'\boldsymbol{T}_\varepsilon\boldsymbol{\varepsilon}=\boldsymbol{D}\boldsymbol{\varepsilon} \tag{7.54}$$

可以证明，$\boldsymbol{T}_\sigma^{-1}=\boldsymbol{T}_\varepsilon^T$，因此

$$\boldsymbol{D}=\boldsymbol{T}_\varepsilon^T\boldsymbol{D}'\boldsymbol{T}_\varepsilon \tag{7.55}$$

除应力应变关系采用上式外，其他所有有限元列式都和空间等参元相同。包括单元刚度阵、质量阵、等效节点力等，这里不再重复给出这些公式。

理论上说，只要采用了(7.55)式的应力应变关系，8～20 节点等参元都可以用于梁板壳的分析。实际上，为确保单元能够模拟挠度在板中面内或沿梁长度方向的变化，除特殊情况外，板上下表面内或梁长度方向的边中节点必须存在。也就是说一般的板壳单元应有 16 个节点，梁单元应有 12 个节点。简单比较可知，16 节点板单元实际上是一种退化的 Mindlin 板单元，20 节点板单元则是退化的符拉索夫板单元。

空间等参单元的退化方法同样适用于平面问题，我们也可以构造 4～8 节点的平面梁、轴对称旋转壳单元等，这里不再给出具体推导，感兴趣的读者完全可以自己完成。

7.3.4　梁板壳单元构造方法总结

以上分析说明，所有构件，如梁、板、壳、膜、弦等都是三维实体的特殊形式。由三维弹性力学结合这些构件的力学特性，引入相应的简化假定，就可以分别得出它们各自的理论。人们根据梁、板、壳等理论按有限元的模式建立了各自的有限单元。由于构造协调的 Kirchhoff 板单元存在困难，于是出现了大量的不协调单元、过协调单元、分片二次插值单元、杂交元等。三维实体等参元是一种简单有效的单元，在分析三维问题时被广泛应用。在三维实体等参元的基础上，直接引入各类构件的简化假定得出了三维退化的各类构件单元（如图 7.9 所示）。

这些实体退化单元不仅与常用的空间等参单元具有相同的自由度数，还能像实体单元一样比较精确地描述结构的几何特性，同时能够按梁、板、壳等结构的变形特征进行有限元分析。实体退化单元没有转角自由度，可以很方便地和其他单元一起用于复杂的组合结构分析。

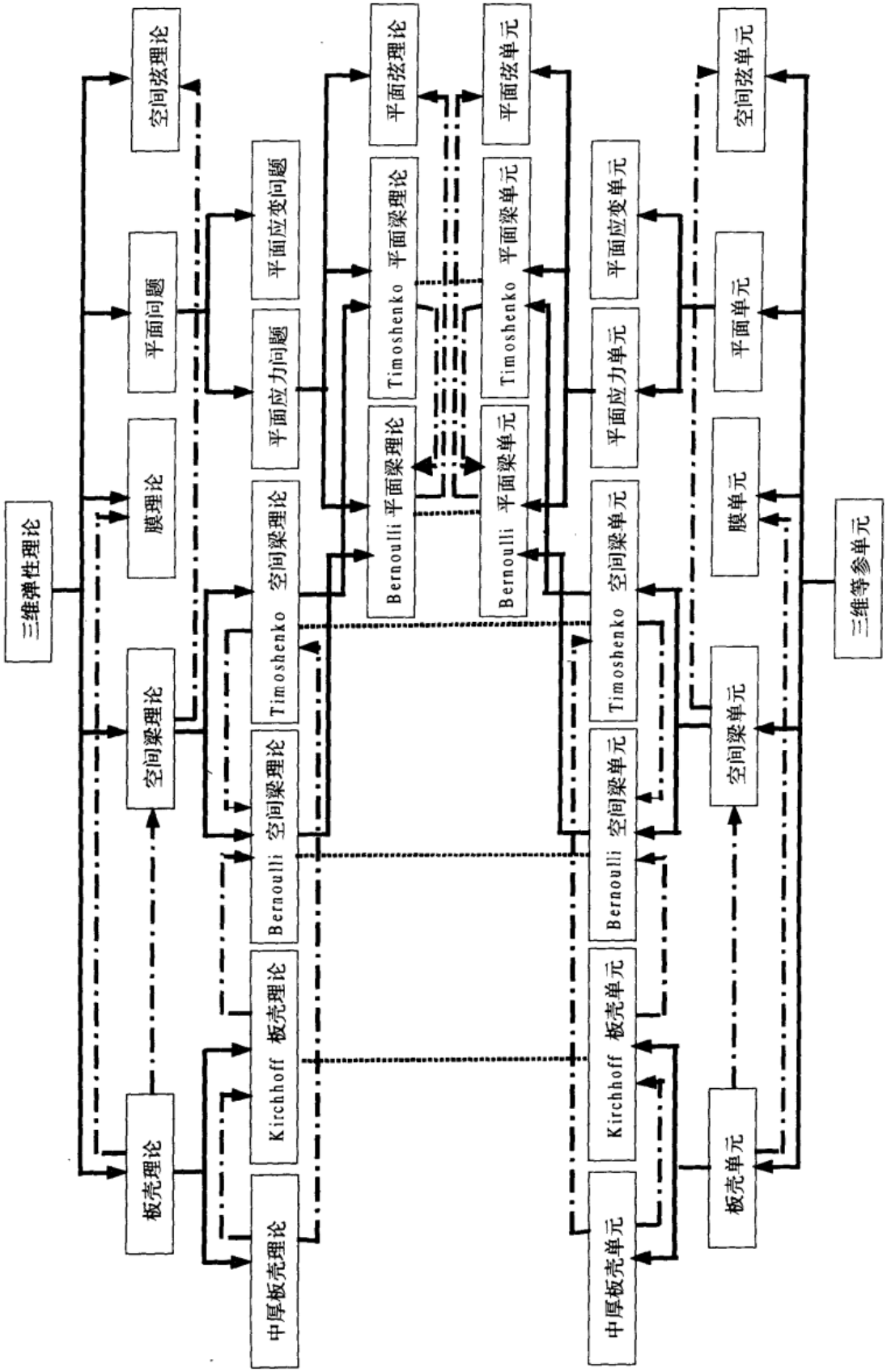


图7.9 三维实体单元和退化单元系列关系

7.3.5 算例

为说明退化单元的适用性和计算精度，我们采用退化的板壳单元分析了一些典型板壳问题，并与解析解及一些经典板壳单元的计算结果进行了比较。

例 7-3 简支和固支方板。

承受均布载荷 q_0 和中心处集中力 p 的四边简支或固支方板，泊松比为 0.3。中心点挠度记为

$$W=\alpha\frac{q_0a^4}{Eh^3}\qquad W=\beta\frac{pa^2}{D}\tag{7.55}$$

式中，a 为边长，h 板厚，α、β为无量纲量，D 为板的弯曲刚度。

表 7.1、7.2 和表 7.3 给出了各种板厚条件下α、β的结果。数值结果表明，实体退化板壳单元的计算精度是令人满意的。对一般工程应用来说，采用较粗的网格求解也已能满足工程实际对精度的要求。

表 7.1　均布荷载作用下简支方板的中心挠度系数α

网格 / α / h/a	2×2	4×4			薄板解[24]
	退化元解	退化元解	文献[25]解	厚板解[27]	
0.01	0.04377	0.04438	0.04438	0.04438	0.04437
0.1	0.04641	0.04628	0.04628	0.05217	
0.2	0.05221	0.05202	0.05202	0.05217	
0.3	0.06179	0.06160	0.06161	0.06192	
0.4	0.07518	0.07500	0.07500	0.07557	

表 7.2　均布荷载作用下固定方板的中心挠度系数α

网格 / α / h/a	2×2	4×4			薄板解[24]
	退化元解	退化元解	文献[25]解	厚板解[27]	
0.01	0.00817	0.01372	0.0137		0.0138
0.1	0.01595	0.01601	0.0160	0.01634	
0.2	0.02214	0.02213	0.0221	0.02331	
0.3	0.03193	0.03195	0.0320		
0.4	0.04544	0.04550	0.0455		

表 7.3　集中荷载作用下固定方板的中心挠度系数 β

网格 / β / h/a	2×2	4×4			薄板解[25]
	退化元解	退化元解	文献[25]解	文献[26]解	
0.01	0.003536	0.005559	0.005559	0.00547	0.00560
0.1	0.006936	0.007301	0.007301	0.00759	
0.2	0.011185	0.012318	0.012318	0.01391	
0.3	0.018118	0.020584	0.020584	0.02436	
0.4	0.027772	0.032102	0.032102	0.03895	

例 7-4 夹持的圆柱壳。

如图 7.10 所示，夹持的圆柱壳，壳厚 h，中心受一对集中力 P 的作用。圆柱两端由横隔板支承，其计算参数如下：

$R=300.0$ $L=600.0$

$E=3.0\times10^6$ $\nu=0.3$

$h=3.0$ $P=1.0$

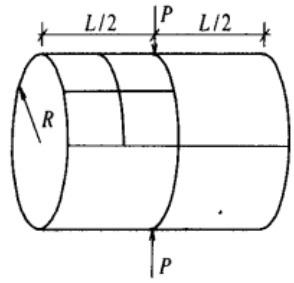


图7.10 夹持的圆柱壳

由于结构和荷载的对称性，只需取 1/8 圆柱壳进行计算。分别采用 2×2、4×4、5×5、6×6 和 8×8 的网格，计算集中力作用处的位移 w。计算结果与有关解的比较列于表 7.4。表中所有其他解均取自文献[28]。为便于比较，所有结果均已关于精确解归一化。集中力作用处位移的精确解[29]为 $w=-0.18248\times10^{-4}$。

表 7.4 横隔板支承圆柱壳标准化解 W_{FEM}/W_{EXACT}

网格	退化元解（16 节点）	4-node SRI	9-node SRI	9-node 3×3	9-node γ-6	9-node γ-4
2×2	0.590	0.373	0.239	0.050	0.743	0.737
4×4	0.898	0.747	0.773	0.161	0.927	0.961
5×5	0.948					
6×6	0.967					
8×8	0.981	0.935	0.856	0.564	0.986	0.999

本节从 8～20 节点三维空间等参元出发，直接引进梁板壳的基本假定，构造了实体退化梁板壳单元。这些单元有以下特点：

一、实体退化单元均是协调单元，单元自由度数均与空间等参单元相同；

二、实体退化单元列式简单划一，程序实现简单方便；

三、各类退化单元间及与空间单元连接十分自然方便；

四、算例说明，退化单元的计算结果是稳定可靠的，无需引入应力应变关系中的系数 λ，对于工程问题来说，退化单元是厚薄板通用的。

7.4 改进的实体退化单元——组合单元

将改进等参元和退化等参元的思想结合，就可以得到非常实用的组合单元。

7.4.1 组合单元

由于退化单元也是等参数单元，因此，可以将改进等参元的思想和退化单元结合，得到改进的退化单元系列。由于这样的单元特别适合层合板壳结构分析，因此有时也称为层合板壳单元。

在桥梁工程中，截面形状复杂（如箱形截面、T 型截面等）的桥梁结构的总体分析是一个常见而又非常繁琐的力学问题。目前对这类结构的分析常采用两种方法。其一是用三维实体单元对桥梁结构进行细致的离散。这一方法的优点是能够准确地描述桥梁的几何形状，它的缺点是庞大的计算量对总体分析而言是一种浪费。其二则是把结构简化为梁、板、壳或梁板

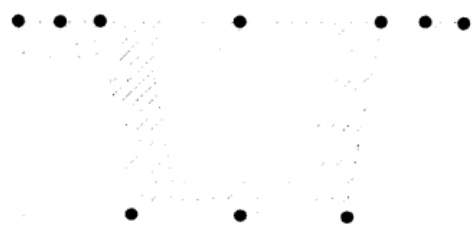


图 7.11 层合单元示意图

壳组合结构。这种方法的优点是计算量少，但很难描述复杂桥梁结构的实际几何特性，特别是在变截面和有中空的区域，因此其结果往往不能反映桥梁的整体特性。如何建立一个既能描述结构的几何形状、又能反映结构受力和变形特征的简单有限元模型是整体分析的关键。为能准确描述几何形状、变形特性，根据改进等参元的特点，可以在结构外设置节点，如图 7.11 中的空心圆点。利用改进的实体退化板壳元，便能建立一个既确切又简单的有限元模型。对于图 7.11 所示截面只需采用三个单元就能较好地描述截面的几何性质，模拟结构的总体变形特点。

改进的实体退化单元在实际应用时要注意两点：第一，退化单元的局部坐标系是按单元中面（$\zeta=0$）为基准定义的，而改进的退化单元中的每个区块需要定义各自的局部坐标系，区块的局部坐标系则以区块中面（$\zeta'=0$）为基准定义。第二，每个区块有独立的局部坐标系和应力应变关系，因此，理论上允许一个单元中包含有一般三维问题、杆、梁、板壳等变形区块，从而构成一个组合单元。这样的组合单元在工程中有很重要的实用价值，已成功应用于桥梁工程和桩筏基础的分析。

采用改进实体退化板壳单元计算了大量的例子[12]~[14]，数值结果表明，单元的计算精度是令人满意的，下面只列出其中的几个。

例 7-5 简支夹层矩形板。

如图 7.12 所示，一简支夹层矩形板表面受均匀分布荷载 q 作用，各层均为正交各向异性材料，上下两个表层的材料相同。材料参数满足如下条件，

$$C_{22}/C_{11}=0.543103 \qquad C_{12}/C_{11}=0.23319$$

$$C_{23}/C_{11}=0.098267 \qquad C_{13}/C_{11}=0.010776$$

$$C_{33}/C_{11}=0.530172 \qquad C_{44}/C_{11}=0.26681$$

$$C_{55}/C_{11}=0.159914 \qquad C_{66}/C_{11}=0.262931$$

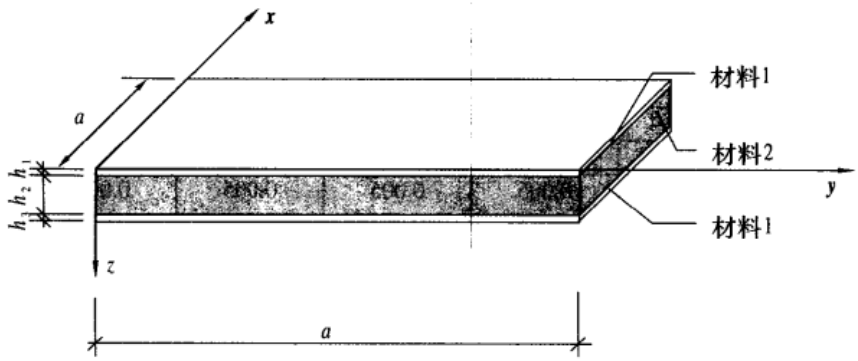


图 7.12　矩形夹层板

记 $\delta=C_{11}^{(1)}/C_{11}^{(2)}$，其中 $C_{11}^{(1)}$ 和 $C_{11}^{(2)}$ 为面层和芯层 C_{11} 的值。当 $\delta=1$ 时，夹心板退化为均质的单层板。采用 4×4×1 的网格计算中心点的挠度 W。计算结果如表 7.5 所示。计算结果表明，改进

的退化板壳单元适合应用于层合结构的有限元分析。

表 7.5 中心点无量纲挠度 $WC_{11}^{(2)}/qh$

δ	J.R.Fang(1996)	S.Srinivas(1970)	L.M.Tang(1992)	退化层合元
1	688.52	688.58	676.05	672.48
10	159.36	159.38	154.18	153.82

例 7-6 悬臂矩形板条。

如图 7.13 所示一悬臂矩形板条，自由端角点处作用一集中荷载 P=10kN。悬臂矩形板条的几何尺寸为 $170\times30\times2\,m^3$，杨氏模量 $E=3.5\times10^{10}$ Pa，泊松比 $\nu=0.2$。采用 $10\times1\times1$ 的有限元网格分析了三种等价情况：真实结构(CASE I)、板条下表面有一 $170\times30\times2\,m^3$ 的虚拟层(CASE II)、板条上表面有一 $170\times30\times2\,m^3$ 的虚拟层(CASE III)。截面 A 处的应力和截面 B 处的挠度计算结果列于表 7.6 和表 7.7 中。

从表可以看出，人为地增加和减少虚拟层不会影响数值结果的精确度。这一点是非常重要的，因为它允许我们根据工程的实际需要增加虚拟层（或区块），以便得到形状规则的网格，从而使得一些本来比较复杂的问题变得非常简单。

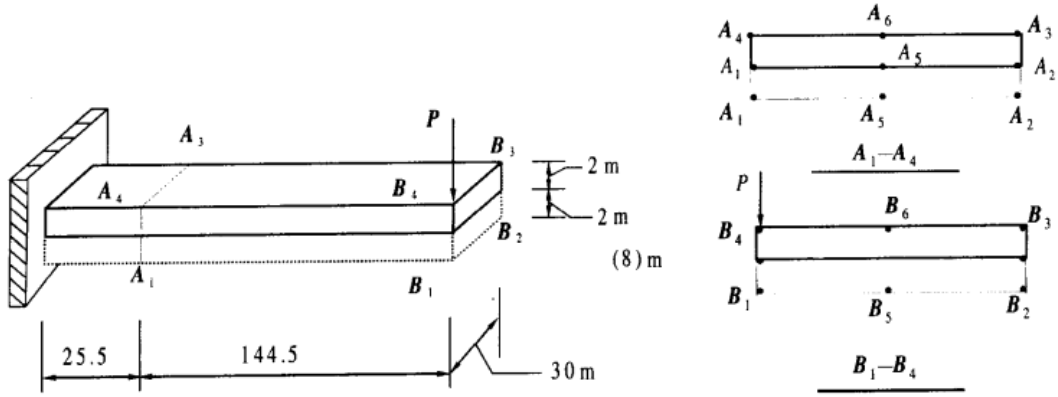


图 7.13 悬臂板条

表 7.6 截面 A_1-A_4 上轴向应力分布 单位:Pa

位 置	$\zeta_1=\zeta_1'$	A_2'	ζ_2	A_4	A_3	A_6
CASE I	-55930.1	-55825.6	-54736.1	55930.1	55825.6	54736.1
CASE II	-55927.4	-55822.7	-54733.4	55927.4	55822.7	54733.4
CASE III	-55921.9	-55817.2	-54728.0	55921.9	55817.2	54728.0
相对误差(%) II(III)	0.005 (0.015)	0.005 (0.015)	0.005 (0.015)	0.005 (0.015)	0.005 (0.015)	0.005 (0.015)

表 7.7 截面 B_1-B_4 上挠度分布 单位:mm

位 置	B_3	B_4	B_6
CASE I	15.9445	16.4406	16.1918
CASE II	15.9437	16.4397	16.1909
CASE III	15.9420	16.4380	16.1892
相对误差(%) II(III)	0.005 (0.016)	0.005 (0.016)	0.005 (0.016)

例 7-7 悬臂箱形薄壁梁。

图 7.14 为一箱形薄壁梁，其杨氏模量 $E=3.5\times10^{10}$Pa，泊松比 $\nu=0.2$。整梁被剖分为 5×1×1 个单元。每个单元分成 8 个区块，其中 4 个遵循板的变形规律，另 4 个（在 4 个角点处）遵循梁的变形规律。数值结果见表 7.8。

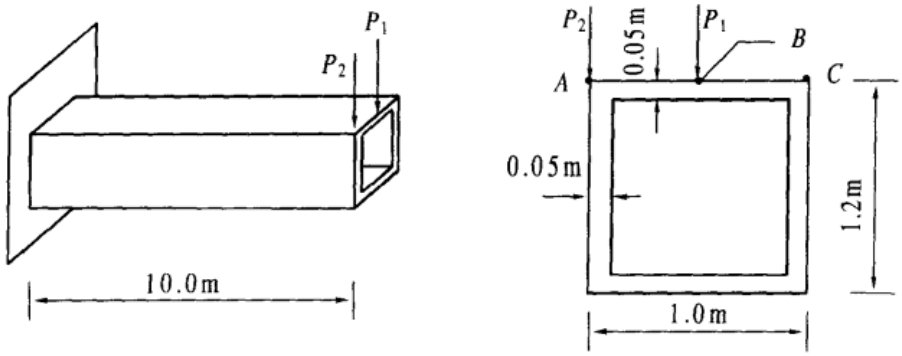


图 7.14　箱形薄壁梁

表 7.8　箱形梁的挠度　　单位：mm

位　置	P_1=10000N　P_2=0N			P_1=0N　P_2=10000N
	Y.Q.Xie(1988)	组合单元	相对误差	组合单元
A	2.156	2.158	0.09%	2.227
B	2.156	2.161	0.23%	2.158
C	2.156	2.158	0.09%	2.089

由表可以看出，虚拟层合单元同样可以应用于分析薄壁结构的弯曲问题。

7.4.2　组合单元中的杆

前面已经指出，一根杆件可以作为组合单元中的一个区块，计算它对组合单元的贡献。这样做的优点是，可以很好地描述复杂杆件的几何形状，能确保和其他区块间位移协调。但对于直的等截面杆件，区块的几何定义有时显得十分繁琐。为此，我们从另一个角度介绍组合单元中的杆，这一思想还可以进一步推广到其他类型的构件，从而大大提高了组合单元的实用性。

空间桁架由只承受拉、压力的杆件组成，通常计算这类结构的有限单元为杆单元。桁架中的每一根杆件作为一个单元。在桁架杆件不多的情形下，杆单元能很好地求解桁架问题。但对于大型结构，如空间网架、大跨度钢桁架桥、用桁架作为加劲梁和主梁的大跨度悬索桥和斜拉桥等，结构中的杆件很多，如果采用杆单元，则形成的总体刚度矩阵很大，影响了解题的速度。曾有作者[32]用“连续化”的平面结构，将高层多跨框架的构件联合起来作为统一整体，然后将这一整体结构沿高度方向划分为几段；在一个段内再划分为几个矩形单元（作者称这一方法为高阶巨型离散元法），以减少计算工作量。本节利用空间 8～20 节点等参单元构造空间桁架单元，把一般杆单元作为空间桁架单元的一个区块。利用空间等参单元的形函数，表示单元内杆端的坐标和位移。这样，通常意义的杆即成了组合单元中的杆。这个单元的优点是显而易见的，因为其未知量数目与桁架中的杆件数无关，从而有效地减少了节点的

数量，缩小了总体刚度矩阵的阶数，提高了解题的速度。由于考虑的是桁架结构变形的整体特性，可以预计它在动力分析、非线性分析中将更能显出其优越性。

一、基本假定

为简化推导，作如下假定：

（1）组合单元中每根杆件都是等截面的，其轴线在整体坐标系中为直线；

（2）组合单元中每根杆件的端点由它在组合单元中的自然坐标确定，其整体坐标和位移由组合单元按位移模式插值确定；

（3）组合单元中每根杆件的位移由该杆件的端点位移线性插值得到。

注意，第一，以上的假定并不是组合单元必需的；第二，当组合单元中有其他类型的构件区块时，假定（3）可能会导致区块间位移不连续。

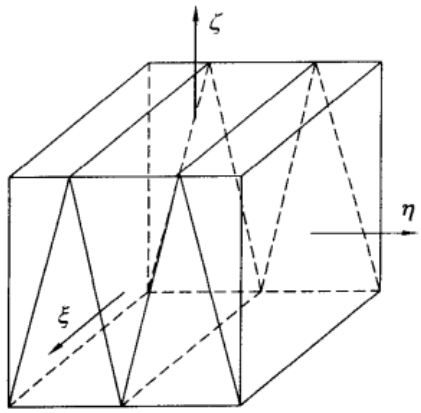


图 7.15 局部坐标系

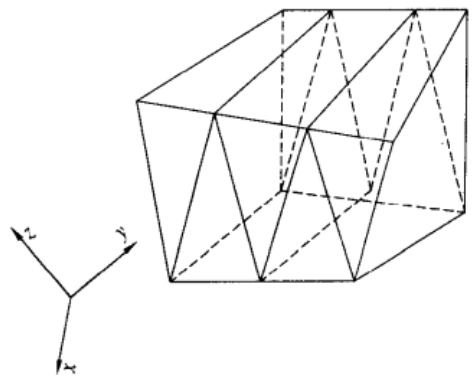


图 7.16 整体坐标系

图 7.15 和 7.16 分别为母单元和组合单元中杆件示意图。现分析组合单元中的一个典型杆件，其端点记为 p 和 q（注意不一定是节点），其坐标和位移分别记为 $\boldsymbol{x}_p$、$\boldsymbol{x}_q$ 和 $\boldsymbol{u}_p$、$\boldsymbol{u}_q$。根据假定，有

$$\boldsymbol{x}_p=\sum_{i=1}^{n}N_i(\xi_p,\eta_p,\zeta_p)\boldsymbol{x}_i \qquad \boldsymbol{x}_q=\sum_{i=1}^{n}N_i(\xi_q,\eta_q,\zeta_q)\boldsymbol{x}_i \tag{7.56}$$

$$\boldsymbol{u}_p=\sum_{i=1}^{n}N_i(\xi_p,\eta_p,\zeta_p)\boldsymbol{u}_i \qquad \boldsymbol{u}_q=\sum_{i=1}^{n}N_i(\xi_q,\eta_q,\zeta_q)\boldsymbol{u}_i \tag{7.57}$$

式中，ξ_p、η_p、ζ_p 和 ξ_q、η_q、ζ_q 分别为端点 p 和 q 的自然坐标。将杆件端点位移写成矢量形式，

$$\boldsymbol{d}^b=\begin{Bmatrix}\boldsymbol{u}_p\\ \boldsymbol{u}_q\end{Bmatrix}=\boldsymbol{N}^b\boldsymbol{d}^e \tag{7.58}$$

式中，

$$\boldsymbol{N}^b=\begin{bmatrix}N_1^p\boldsymbol{I}_{3\times3} & N_2^p\boldsymbol{I}_{3\times3} & \cdots & N_n^p\boldsymbol{I}_{3\times3}\\ N_1^q\boldsymbol{I}_{3\times3} & N_2^q\boldsymbol{I}_{3\times3} & \cdots & N_n^q\boldsymbol{I}_{3\times3}\end{bmatrix} \tag{7.59}$$

式中，$N_i^p=N_i(\xi_p,\eta_p,\zeta_p)$，$N_i^q=N_i(\xi_q,\eta_q,\zeta_q)$。

二、杆件的应力、应变和轴力

设杆件轴线与 x、y 和 z 轴夹角的余弦分别为 $\cos\alpha$、$\cos\beta$ 和 $\cos\gamma$。对于小应变问题，

不难得到杆的应变为

$$\varepsilon=\frac{\cos\alpha}{l}(u_q-u_p)+\frac{\cos\beta}{l}(v_q-v_p)+\frac{\cos\gamma}{l}(w_q-w_p) \tag{7.60}$$

则杆的应力和轴力分别为

$$\sigma=E\varepsilon \qquad N=\sigma A \tag{7.61}$$

式中，l 为单元的长度，E 为杆件的拉伸模量。

三、单元刚度矩阵

如果记

$$\boldsymbol{T}^b=\begin{bmatrix}\cos^2\alpha & \cos\alpha\cos\beta & \cos\alpha\cos\gamma\\ \cos\alpha\cos\beta & \cos^2\beta & \cos\beta\cos\gamma\\ \cos\alpha\cos\gamma & \cos\beta\cos\gamma & \cos^2\gamma\end{bmatrix} \tag{7.62}$$

则容易导出杆件的刚度矩阵为

$$\boldsymbol{K}^b=\frac{EA}{l}\begin{bmatrix}\boldsymbol{T}^b & -\boldsymbol{T}^b\\ -\boldsymbol{T}^b & \boldsymbol{T}^b\end{bmatrix} \tag{7.63}$$

杆端力 $\boldsymbol{F}^b$ 与杆端位移 $\mathbf{d}^b$ 之间满足如下关系式：

$$\boldsymbol{F}^b=\boldsymbol{K}^b\boldsymbol{d}^b \tag{7.64}$$

将式(7.58)代入上式，然后在方程两边同时乘以 $\boldsymbol{T}^b$ 的转置，即可得到该杆件对组合单元刚度矩阵的贡献，

$$\boldsymbol{K}=(\boldsymbol{N}^b)^T\boldsymbol{K}^b\boldsymbol{N}^b \tag{7.65}$$

同样，我们可以得到杆件端点力 $\boldsymbol{F}^b$ 对组合单元等效节点力的贡献，

$$\boldsymbol{F}=(\boldsymbol{N}^b)^T\boldsymbol{F}^b \tag{7.66}$$

以上讨论的内容是针对一根典型杆件的，如果组合单元中有多根杆件，只需依次计算每根杆件对组合单元的贡献，并累加即可。

例 7-8 简支桁架梁（一）。

某一简支桁架梁，由长度均为 1.9m 的等边三角形组成，结构形式如图 7.17，杆件弹性模量 $E=2.1\times10^5$MPa，各杆截面积：$A_1=18\text{cm}^2, A_2=43\text{cm}^2$，$A_3\sim A_6=31\text{cm}^2$，$A_7=61\ \text{cm}^2$，$A_8=31\ \text{cm}^2$，$A_9=69\ \text{cm}^2$。8 号节点受到一向下集中力 P=900kN 的作用。

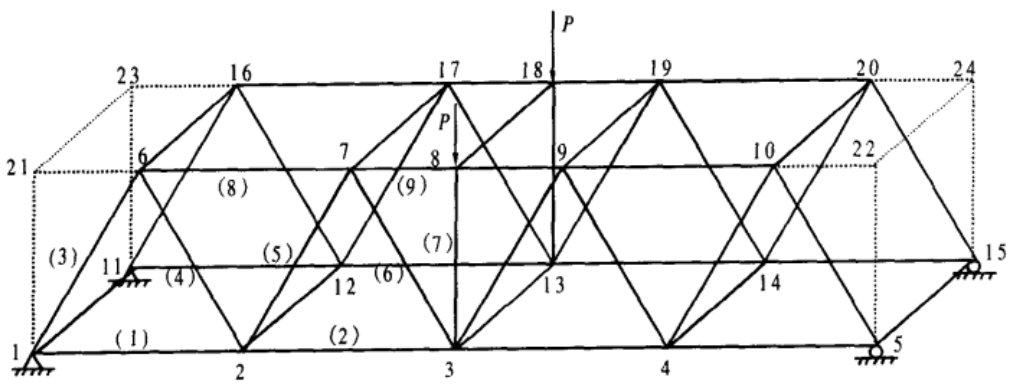


图 7.17　简支桁架梁（一）节点、单元编号示意图

分别用杆单元和空间桁架 12 节点单元进行静力计算。对于杆单元，各节点间的杆件作为一个单元，而对于空间桁架 12 节点单元可以有多种划分方式，本例仅列出两种。第一种划分方式是由节点 11、13、18、16、1、3、8、6、12、17、7、2 和 13、15、20、18、3、5、10、8、14、19、9、4 组成两个单元。第二种划分方式也将此结构分为两个单元，即节点 11、13、18、23、1、3、8、21、12、17、7、2 和 13、15、24、18、3、5、22、8、14、19、9、4 组成两个单元（21～24 节点为虚拟的节点）。对于两个单元公共边界上的杆件，可将其划给其中一个单元，也可将其截面积一分为二后分给相邻的两个单元。

计算结果表明，两种单元划分方式虽然不同，但计算结果却是一样的。杆单元和空间桁架单元的各杆内力计算结果列于表 7.9 中，各节点位移列于表 7.10 中。从两个表中数值可以看出，组合单元和杆单元的计算结果是一样的。

表 7.9 组合单元计算结果与杆单元计算结果（轴力）的比较

杆号	例7-8		例7-9		
	组合单元(kN)	杆单元 (kN)	组合单元 (kN)	杆单元 (kN)	误 差 (%)
(1)	259.81	259.81	259.23	259.81	–0.22
(2)	779.42	779.42	285.15	259.81	9.75
(3)	–519.62	–519.62	743.13	779.42	–4.66
(4)	519.62	519.62	805.06	779.42	3.29
(5)	–519.62	–519.62	–522.42	–519.62	–0.54
(6)	519.62	519.62	–2.22	0.00	
(7)	900.00	900.00	502.69	519.62	–3.26
(8)	–519.61	–519.62	57.90	0.00	
(9)	1039.23	1039.23	–545.04	–519.62	–4.89
(10)			–26.09	0.00	
(11)			508.30	519.62	–2.18
(12)			–861.40	–900.00	4.29
(13)			–520.93	–519.62	–0.25
(14)			–495.80	–519.62	4.58
(15)			–1047.60	–1039.23	–0.81

例 7-9 简支桁架梁（二）。

结构形式如图 7.18 所示，与图 7.17 相比较增加了竖杆，竖杆的材料与其他杆件一样，截面积 $A=18\ \text{cm}^2$，对于杆单元，仍然取各节点间的杆件作为一个单元，而对于空间桁架 12 节点单元，还是将此结构分为两个单元，即 17、21、29、26、1、5、13、10、19、27、11、3 和 21、25、32、29、5、9、16、13、23、31、15、7 两个单元。

各杆内力计算结果列于表 7.9 中，各节点位移见表 7.10。从表中数值可以看出，两者的计算结果有误差。但误差均小于 10%。产生误差的原因是采用空间桁架单元节点数少，仅为采用杆单元节点数的 62.5%。考虑到误差值并不大，而且空间桁架单元的节点数少于杆单元，形成总体刚度矩阵的阶数也相应较小，说明空间桁架单元具有一定的优越性。

表 7.10　组合单元计算结果与杆单元计算结果（节点位移）的比较

节点号	例7-8				例7-9			
	组合单元计算结果		杆单元计算结果		组合单元计算结果		杆单元计算结果	
	u (mm)	v (mm)	u (mm)	v (mm)	u (mm)	v (mm)	u (mm)	v (mm)
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.31	-8.69	1.31	-8.69	/	/	0.65	-4.72
3	2.95	-13.92	2.95	-13.92	1.37	-8.63	1.31	-8.69
4	4.59	-8.69	4.59	-8.69	/	/	2.13	-11.78
5	5.89	0.00	5.89	0.00	3.00	-13.92	2.95	-13.92
6	5.14	-4.72	5.14	-4.72	/	/	3.77	-11.78
7	3.99	-10.07	3.99	-10.07	4.63	-8.63	4.59	-8.69
8	2.95	-15.08	2.95	-15.08	/	/	5.24	-4.72
9	1.90	-10.07	1.90	-10.07	5.99	0.00	5.89	0.00
10	0.75	-4.72	0.75	-4.72	5.17	-4.74	5.14	-4.72
11					4.41	-8.38	4.39	-8.69
12					/	/	3.63	-11.78
13					3.00	-15.02	2.95	-15.08
14					/	/	2.26	-11.78
15					1.59	-8.38	1.51	-8.69
16					0.83	-4.74	0.75	-4.72

注：w 的值为零，表中未列出。

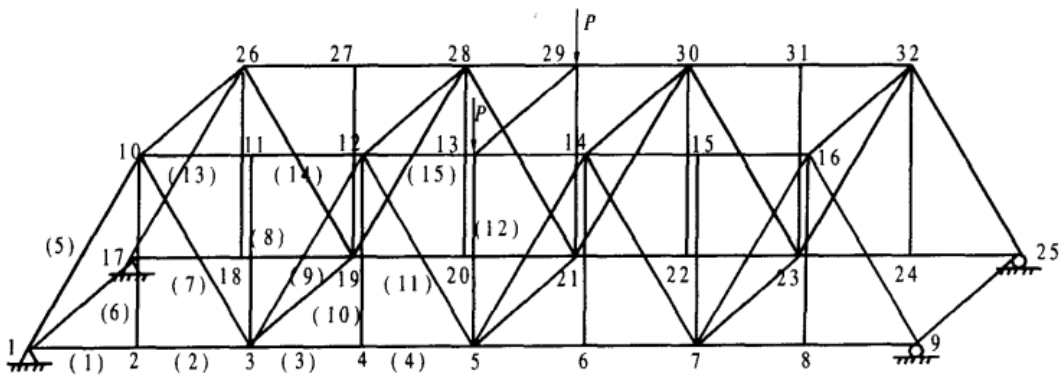


图 7.18　简支桁架梁（二）节点、单元编号示意图

7.4.3　组合单元中的钢筋

在钢筋混凝土和预应力钢筋混凝土结构中，作为构造钢筋的分布钢筋网可以作为单元模型中的一层来体现它们对单元刚度矩阵的贡献[3]。但在结构中，还有作为主要受力钢筋的粗大钢筋或预应力筋束。对这些钢筋的处理，通常是在划分单元时作为单独的单元依靠节点连接在相应的混凝土单元上。由于结构中这样的钢筋很多，这样处理必然导致细小单元的划分，计算工作量大大增加。本节把这些钢筋作为组合单元中的一部分，使其与周围混凝土共同协调作用，并将它们在结构中的作用转换为对所在单元刚度矩阵的贡献。文献[33]也采用了这种思想，但是其中钢筋的形状有限制，如空间问题假定某根钢筋局部坐标 ξ、η、ζ 中的两个必须是常数，故不能描述弯起筋或曲线筋。

一、基本假设

（1）假设钢筋与周围混凝土完全粘结，没有任何相对变位。即钢筋应变等于其周围混凝土的应变；

（2）钢筋只承受轴向力。

二、钢筋的几何描述

如图 7.19 所示，在某单元中有一根曲线钢筋。其端点及中点分别为点 1、2、3。设三点的自然坐标为 (ξ_i,η_i,ζ_i)，$(i=1,2,3)$，则钢筋上任意一点的自然坐标可表示为

$$\xi=\sum_{i=1}^{3}l_i(t)\xi_i \quad \eta=\sum_{i=1}^{3}l_i(t)\eta_i \quad \zeta=\sum_{i=1}^{3}l_i(t)\zeta_i \tag{7.67}$$

其中，$t\in[-1,1]$，$l_i(t)$ 和（6.48）～（6.50）式中的 $N_i(\xi)$ 具有完全相同的形式。

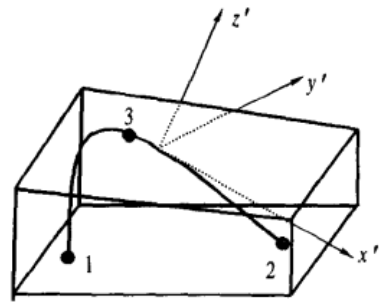


图 7.19 曲线钢筋形状描述

三、钢筋的应力、应变

根据假设（1），钢筋的应变可以按照周围混凝土的应变计算，而单元中任一点的应变都是该点自然坐标（ξ,η,ζ）的函数。因此，利用（7.67）式总可以求得钢筋中任意点的轴向应变 ε。但要注意的是，组合单元的应变有 6 个分量，并非就是钢筋轴向应变。为了计算钢筋的轴向应变，必须先求得钢筋内任意点的单位轴线方向矢量 $\{l,m,n\}^T$，

$$l=\frac{\mathrm{d}x}{\mathrm{d}s}=\frac{\mathrm{d}x/\mathrm{d}t}{\sqrt{(\mathrm{d}x/\mathrm{d}t)^2+(\mathrm{d}y/\mathrm{d}t)^2+(\mathrm{d}z/\mathrm{d}t)^2}} \tag{7.68}$$

$$m=\frac{\mathrm{d}y}{\mathrm{d}s}=\frac{\mathrm{d}y/\mathrm{d}t}{\sqrt{(\mathrm{d}x/\mathrm{d}t)^2+(\mathrm{d}y/\mathrm{d}t)^2+(\mathrm{d}z/\mathrm{d}t)^2}} \tag{7.69}$$

$$n=\frac{\mathrm{d}z}{\mathrm{d}s}=\frac{\mathrm{d}z/\mathrm{d}t}{\sqrt{(\mathrm{d}x/\mathrm{d}t)^2+(\mathrm{d}y/\mathrm{d}t)^2+(\mathrm{d}z/\mathrm{d}t)^2}} \tag{7.70}$$

式中，

$$\begin{aligned}
\frac{\mathrm{d}x}{\mathrm{d}t}&=\frac{\partial x}{\partial\xi}\frac{\partial\xi}{\partial t}+\frac{\partial x}{\partial\eta}\frac{\partial\eta}{\partial t}+\frac{\partial x}{\partial\zeta}\frac{\partial\zeta}{\partial t}\\
\frac{\mathrm{d}y}{\mathrm{d}t}&=\frac{\partial y}{\partial\xi}\frac{\partial\xi}{\partial t}+\frac{\partial y}{\partial\eta}\frac{\partial\eta}{\partial t}+\frac{\partial y}{\partial\zeta}\frac{\partial\zeta}{\partial t}\\
\frac{\mathrm{d}z}{\mathrm{d}t}&=\frac{\partial z}{\partial\xi}\frac{\partial\xi}{\partial t}+\frac{\partial z}{\partial\eta}\frac{\partial\eta}{\partial t}+\frac{\partial z}{\partial\zeta}\frac{\partial\zeta}{\partial t}
\end{aligned} \tag{7.71}$$

于是，钢筋轴线方向的应变为

$$\varepsilon=\varepsilon_x l^2+\varepsilon_y m^2+\varepsilon_z n^2+\gamma_{yz}mn+\gamma_{zx}nl+\gamma_{xy}lm=\boldsymbol{T}\,\boldsymbol{\varepsilon} \tag{7.72}$$

钢筋中轴向应力为

$$\sigma=E\varepsilon \tag{7.73}$$

四、钢筋对组合单元刚度矩阵的贡献

根据以上的推导，组合单元中任意一根钢筋对组合单元刚度矩阵的贡献可以表示为

$$\boldsymbol{K}=EA\int_l \boldsymbol{B}^T\boldsymbol{T}^T\,\boldsymbol{T}\,\boldsymbol{B}\,\mathrm{d}s \tag{7.74}$$

或

$$\boldsymbol{K}=EA\int_{-1}^{1}\boldsymbol{B}^{T}\boldsymbol{T}^{T}\boldsymbol{T}\boldsymbol{B}\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^{2}+\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^{2}+\left(\frac{\mathrm{d}z}{\mathrm{d}t}\right)^{2}}\,\mathrm{d}t \tag{7.75}$$

式中，A 是钢筋的面积，l 为钢筋的长度。式（7.76）中，积分符号内各项都用 t 表示，很容易转化为对 t 的高斯积分。

同理，一根典型钢筋对单元质量矩阵的贡献可表示为

$$\boldsymbol{M}=\int_{-1}^{1}\rho A\boldsymbol{N}^{T}\boldsymbol{N}\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^{2}+\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^{2}+\left(\frac{\mathrm{d}z}{\mathrm{d}t}\right)^{2}}\,\mathrm{d}t \tag{7.76}$$

式中，ρ 为钢筋的质量密度。

7.5　组合单元的应用

改进的实体退化梁板壳单元已被广泛地应用于桥梁工程、航空航天工程，均获得了很好的结果。这里只给出其中的一部分。

7.5.1　组合单元在约束扭转问题中的应用

组合单元不仅可以分析梁的弯曲问题，也可以分析杆件的约束扭转问题。一般地说，约束扭转问题的截面翘曲函数是很复杂的，如果一个截面离散为一个单元，便无法表示出截面翘曲的变化。例如，对于正方形截面杆的约束扭转问题，有限元分析时应取截面的 1/4 为一单元，而且截面上应至少有 8 个节点。

一、实心杆件的扭转问题

考虑一端固定（不能转动，但轴向自由），一端受扭矩作用，截面分别为椭圆（长轴 2m、短轴 1m；杆长 30m，扭矩 $T=8\times10^{5}$Nm）、正三角形（边长 1m;杆长 10m,自由端沿边界作用 $q=10^{5}$N/m 的切向线分布力）、正方形(边长 0.5m,杆长 10m,自由端沿边界作用 $q=10^{5}$N/m 的切向线分布力)和长方形（边长 1.0m、宽 0.5m;杆长 10m,自由端沿边界作用 $q=10^{5}$N/m 的切向线分布力）的实心杆的扭转。材质均为 E=210GPa，ν=0.25。杆沿横截面对称轴分别划分成 4、3、4、4 等份(四边形)，根据结构的对称性和载荷的反对称性取其中的一份进行分析,沿轴向划分为 5 个单元。自由端截面上一点的位移计算结果与理论解[30]比较如表 7.11,其中 u 为轴向位移。

表 7.11　实心杆自由端位移比较　　单位：10^{-5} m

截面类型	椭圆截面（-1，$\sqrt{3}/2$）			正三角形（$-\sqrt{3}/2$，-1）		
	u	v	w	u	v	w
退化梁元	0.102	2.481	-2.865	0.131	1.947	-1.685
弹性理论	0.099	2.461	-2.842	0.129	1.984	-1.718
相对误差	3.7%	0.8%	0.8%	1.5%	1.8%	1.9%
截面类型	正方形（最大翘曲点）			长方形（最大翘曲点）		
	u	v	w	u	v	w
退化梁元	0.5944	2.077	-4.189	2.416	5.116	-2.558
弹性理论	0.5889	2.116	-4.233	2.427	5.198	-2.599
相对误差	0.9%	1.8%	1.0%	0.45%	1.6%	1.6%

二、悬臂矩形箱梁的约束扭转问题

考虑尺寸如图 7.20 所示的悬臂矩形箱梁的约束扭转。材料弹性常数为 $E=2.1\times10^7 N/cm^2$，$\nu=0.25$，自由端受纯扭转荷载作用。利用结构的对称性及荷载的反对称性，取 1/4 结构计算。采用两种不同的模型分析。梁板组合模型把构件沿轴向划分为 20 等份，横截面分三部分，共 40 个三维退化板单元，20 个三维退化梁单元，538 个节点；三维块体模型把构件沿轴向分 50 等份，横截面内划分成 10 部分，共 500 个单元，3803 个节点。

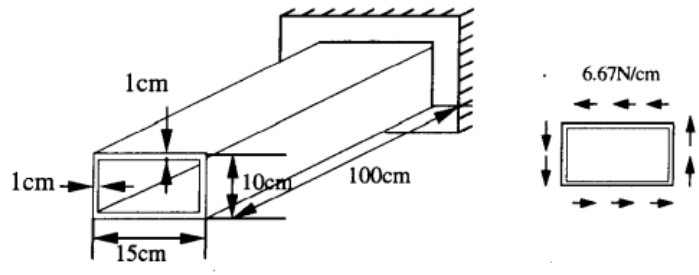


图 7.20 悬臂矩形箱梁

表 7.12 两种模型的自由端角点翘曲位移比较 单位：cm

位移分量	u(10^{-6})	v(10^{-5})	w(10^{-4})
梁板组合模型	2.689	7.869	1.282
三维块体模型	2.511	8.198	1.304
相对误差（%）	6.7	3.2	2.5

表 7.13 两种模型的自由端角点翘曲正应力比较 单位：N/cm^2

截面位置	三维块体模型	梁板组合模型	广义坐标法	折板壳单元[31]
0.025L	1.959	1.830	1.251	1.217
0.075L	1.491	1.418	0.973	0.940
0.125L	1.027	1.018	0.722	0.686
0.175L	0.649	0.661	0.507	0.473
0.225L	0.359	0.388	0.319	0.303

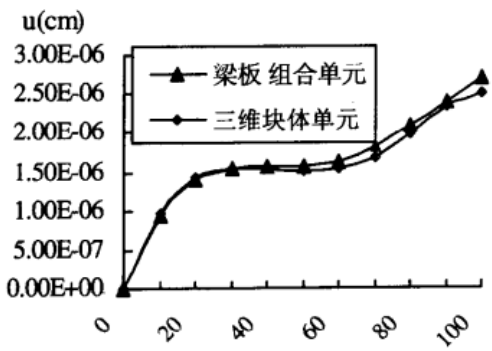


图 7.21 角点翘曲位移沿轴向变化曲线

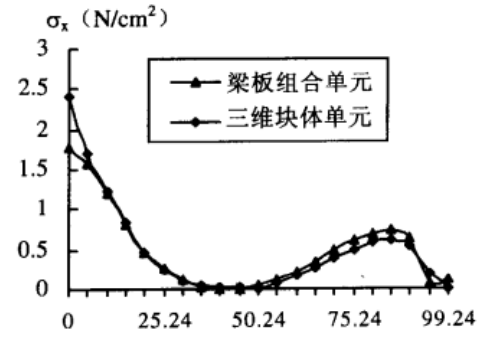


图 7.22 角点翘曲正应力沿轴向变化曲线

表 7.12 给出了两种模型的自由端角点翘曲位移的比较，表 7.13 中的翘曲正应力是指腹板和翼缘交接的折线区域中心点处的翘曲正应力σ_x，图 7.21 和图 7.22 分别为角点处的翘曲位移和翘曲正应力沿轴向的变化曲线。其中表 7.13 中的截面位置指距离固定端的长度。由图可见，两种模型计算得到的位移吻合得很好。精细的三维块体模型表明，箱梁腹板与翼缘交接区域的应力状态比较复杂，沿轴向变化较快。除固定端以外，梁板组合模型具有很好的计算精度。

表 7.13 则表明采用梁板组合模型计算的翘曲正应力值比广义坐标法和折板壳单元法具有更高的精度，充分说明了组合单元的适用性和精确性。

7.5.2 复合材料层合板的振动

图 7.23 为一矩形层合板，长 750mm，宽 600mm，四边自由。沿厚度方向分为五层，各层材料依次为：(一)铝板层，弹性模量为 $E=6.89\times10^{10}$Pa，泊松比 $\nu=0.34$，厚度为 1mm；(二)阻尼层，各向同性材料，假定为开尔文模型，$\eta=1.72$N/mm^2，厚度为 3mm；(三)碳纤维复合材料层，碳纤维采用（±45/0/90/0/∓45）对称铺设，每层厚度 0.125mm，为正交各向异性材料（航天材料，参数从略）；(四)蜂窝材料层，假定该层仅承受横向剪切作用，厚度为 68.25mm（航天材料，参数从略）；(五)碳纤维复合材料层，同第三层。

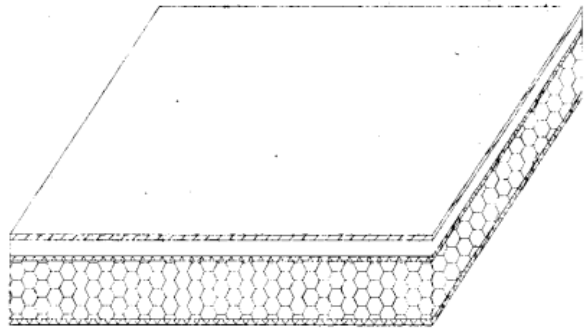


图 7.23　具有约束阻尼层的蜂窝夹层板示意图

其中第一、二层合称为约束阻尼层。无约束阻尼层的蜂窝夹层板和有约束阻尼层的蜂窝夹层板自由振动频率的计算结果和实测结果分别如表 7.14、7.15 所示，表中空白格为实验未测得频率。

表 7.14　无约束阻尼层蜂窝夹层板自由振动频率　　单位：Hz

阶次	1	2	3	4	5	6	7	8	9	10	11
FEM	425.6	692.9	732.9	747.5	909.9	961.1	1036.1	1089.9	1452,0	1545.2	1600.0
实验	433.6	598.7			905.5	930.5	1023.5		1542.3	1318.2	1579.4

表 7.15　有约束阻尼层蜂窝夹层板自由振动频率　　单位：Hz

阶次	1	2	3	4	5	6
FEM	390.1	617.5	849.22	890.0	948.0	980.0
实验	399.3	539.3	845.0	829.5		

由表 7.14、7.15 可见，计算结果与实验结果吻合得很好。组合单元应用于工程实践的一个最大的优点就是，在保持同等精度的前提下效率非常高，这是由于它与分层单元相比减少了大量的节点，从而减少了计算量，节约了机时。这一点对动力问题和非线性问题的分析尤为有利，使得很多层合板壳问题的微机分析成为可能。对上述工程实例，用分层单元计算所用机时将增加 20 倍之多，可见对更加复杂的动力问题，分层单元几乎是无能为力的。

7.5.3 形状优化[12]

结构形状优化问题常常涉及边界形状的改变，因此常常需要反复生成有限元网格，并且

可能由于出现病态网格导致整个优化失败。利用虚拟曲线层合单元进行结构形状优化，把特定的层厚作为设计变量，无须改变原始数据结构，因此，避免了优化过程中有限元网格的反复生成。算例表明此法计算简洁，结果可靠，并有良好的工程应用前景。

如图 7.24 所示悬臂梁，梁长 $l=6.0\,\text{cm}$，梁高 $h_0=1.0\,\text{cm}$，梁宽 $w=1.0\,\text{cm}$，即图 7.24 中粗实线所示。弹性模量 $E=30000\ \text{N/cm}^2$，集中力 $P=1.0\,\text{N}$。

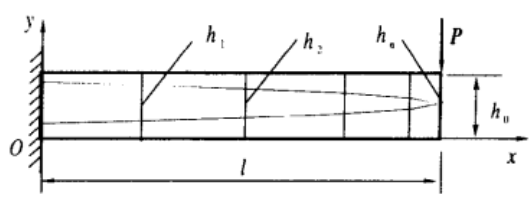


图7.24 悬臂梁

将梁分为三层，细实线所围为中间层，弹性模量 $E=30000\ \text{N/cm}^2$，上下二层则取 $E=0$。h_j 为设计变量，变量个数为 5。取中心层的面积 S 为优化目标，约束条件为 $(\sigma_x)_{\max}\le[\sigma]$，其中，$[\sigma]$ 为许用应力，取 $[\sigma]=60.0\,\text{N/cm}^2$。于是，形状优化问题归结为如下非线性规划问题，

$$\begin{aligned}&\min S(\boldsymbol{H})\ , && \boldsymbol{H}\in E^n\\ &\text{s.t.}\ \ (\sigma_x)_{\max}-[\sigma]\le 0 &&\\ &h_j\ge 0 && j=1,2,\cdots,n\end{aligned}$$

式中，$\boldsymbol{H}=\{h_1,h_2,\cdots,h_n\}^T$。

形状优化结果与按等强度理论解析解 $h=\sqrt{\dfrac{6(l-x)}{w[\sigma]}}$ 的比较见表 7.16。图 7.24 中细实线所示即为优化形状。

表 7.16 悬臂梁形状优化结果 单位：cm

x	0.0	1.5	3.0	4.5	5.6
解析解 h_j	0.775	0.671	0.548	0.387	0.200
有限元解 h_j	0.776	0.672	0.551	0.393	0.213
相对误差	0.1%	0.15%	0.5%	1.6%	6.5%

如图 7.25 所示为一简支梁，长 $l=12.0\,\text{cm}$，高 $h_0=1.0\,\text{cm}$，宽 $w=1.0\,\text{cm}$。弹性模量 $E=30000\ \text{N/cm}^2$，分布力 $q=1.0\ \text{N/cm}$。取许用应力 $[\sigma]=60.0\,\text{N/cm}^2$。

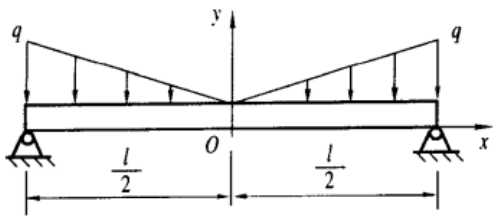


图 7.25 简支梁

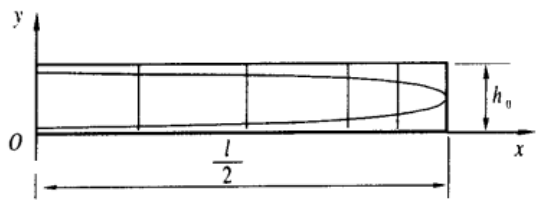


图 7.26 简支梁优化结果

按等强度理论，可以推出梁的高度如下式变化，

$$h=\sqrt{\frac{\frac{ql^2}{4}\left[1-\left(\frac{2x}{l}\right)^3\right]}{w[\sigma]}}$$

利用梁的对称性，取一半进行优化分析，形状优化结果与解析解的比较见表 7.17。优化形状为图 7.26 的细实线部分。

表 7.17　简支梁形状优化结果　　单位：cm

x	0.00	1.50	3.00	4.50	5.25
解析解 h_j	0.775	0.769	0.725	0.589	0.445
有限元解 h_j	0.776	0.774	0.732	0.605	0.448
相对误差	0.1%	0.7%	1.0%	2.7%	0.7%

参考文献

[1] X.Xu, R.F.Cai. A new plate shell element of 16 nodes 40 degrees of freedom by relative displacement method. *Communications in Numerical Methods in Engineering*, 1993, 9

[2] 徐兴，凌道盛.实体退化单元系列.固体力学学报，2001，22（计算力学专辑）

[3] 凌道盛，张金江，项贻强，徐兴.虚拟层合板单元及其在桥梁工程中的应用.土木工程学报，1998, 31(3)

[4] 凌道盛.有自有面渗流分析的虚节点法.浙江大学学报（工学版），2002，36（3）

[5] 凌道盛，陈云敏，丁皓江.任意基础板的有限元分析，*岩土工程学报*，2000，22（4）

[6] 蔡金标，凌道盛，徐兴.大跨度悬索桥振动分析的组合单元法.中国公路学报，2003，16（4）

[7] 程晓东，徐兴，凌道盛.钢管混泥土轴心受压构件极限承载力的有限元分析.固体力学学报，2003, 23(4)

[8] 凌道盛，李芳，徐兴.多功能平面单元及其应用.固体力学学报，2001，22（计算力学专辑）

[9] 丁利，凌道盛，王立忠，陈云敏.组合单元法在砂井地基有限元分析中的应用.计算力学学报，2004, 21 (1)

[10] Xu Xing, Li Fang, Ling Daosheng. Application of virtual laminated element in topology optimization of structure.*Journal of Zhejiang University(SCIENCE)*，2001, 2(2)

[11]Li Fang, Xu Xing, Ling Daosheng. Shape optimization of plate with static and dynamic constraints via virtual laminated element.*Journal of Zhejiang University(SCIENCE)*, 2003, 4(2)

[12]李芳.结构形状和拓扑优化的虚拟层合单元法，浙江大学博士学位论文，2001。

[13]叶贵如.斜拉桥非线性有限元计算和施工过程分析.浙江大学博士学位论文，2000。

[14]蔡金标.大跨度悬索桥空间分析的组合单元法.浙江大学博士学位论文，2002

[15]仵彦卿.岩体水利学导论. 成都:成都西南交通大学出版社，1994

[16]薛禹群，谢春红.水利地质学的数值方法.北京:煤炭工业出版社，1980

[17]吴梦喜，张学勤.有自由面渗流分析的虚单元法.水利学报，1994，(8)

[18]J.N. Bathe. Transmit matrix method for seepage with free surface problem. *Int. J. Num. Meth. Engng*, 1983, (7)

[19]李春华.稳定渗流有限元计算时采用固定网格法的初步研究，第三届全国渗流力学讨论会论文汇编。

[20]王贤能等.有自由面渗流分析的高斯点法.水文地质工程地质, 1997，(6)

[21]C.S. Desai. Finite element residual schemes for unconfind flow. *Int. J. Num. Meth. Engng.*, 1976，(10)

[22]张有天等.有自由面渗流分析的初流量法.水利学报，1988，(8)

[23]梁业国等.有自由面渗流分析的子单元法.水利学报，1997，(8)

[24]S. 铁摩辛柯，S.沃诺斯基，《板壳理论》翻译组译.板壳理论. 北京:科学出版社，1977

[25]谢贻权,何福保.弹性和塑性力学中的有限单元法. 北京:机械工业出版社，1981

[26]周卫宇.平板弯曲问题的数值方法研究.浙江大学硕士论文，1984

[27]S.Rinivas, A.K.Rao, Flexure of thick rectangular plates, *J.A.M.*, 1973, 40(1).

[28]T.Belytschko, B.L.Nong. and H.Stolarski. Assumed strain stabilization procedure for the 9-node Lagrange shell element. *Int. J. Numer. Methods. Eng.* 1989, 28

[29]E.N. Doorkin and K.J.Bathe. A continues mechanics based four-node shell element for general nonlinear analysis. *Eng. Comp.* 1984, (1)

[30]王启德.应用弹性理论. 北京:机械工业出版社，1966

[31]胡毓仁，陈伯其.一种新的薄壁杆单元扭转刚度矩阵.计算结构力学及其应用，1988，5（3）

[32]周　岱，刘红玉. 高层多跨框架计算的高阶巨型离散元法. 土木工程学报，1994，(6)

[33]朱伯芳.有限单元法原理与应用. 北京:中国水利水电出版社, 1998

第8章

大变形问题的有限单元法

以上各章我们介绍了几何非线性问题的变形、应变和应力的性质和描述方法，并建立起积分形式的平衡方程，还进一步介绍了线性问题的有限元法，为本章建立几何非线性有限元方程奠定了必要的基础。由于涉及的问题众多，几何非线性问题的种类亦十分繁杂，不可能在一章中全面加以介绍，因此，本章以三维问题的等参单元为例详细地阐述几何非线性问题的有限元分析方法，并介绍几何非线性杆单元和梁单元。

8.1 有限元法方程的建立——Lagrange 法

以初始构形为参考构形，将初始构形进行有限单元离散，选用固定不动的直角坐标系。初始构形中单元的几何形状由单元的节点坐标插值得到，

$$\boldsymbol{X}=\sum_{k=1}^{n}N_k(\xi,\eta,\zeta)\boldsymbol{X}_k \tag{8.1}$$

式中，$\boldsymbol{X}=\{X_1,X_2,X_3\}^T$ 为任意物质点的坐标矢量，$\boldsymbol{X}_k$ 为第 k 个节点的物质坐标，$N_i(\xi,\eta,\zeta)$ 为定义在母单元上的插值形函数，其形式与线性有限元法中的插值形函数完全相同。注意式(8.1)和(6.88)式形式上是相似的，但实际上是有区别的。单元中任意一点的位移可以表示为

$$\boldsymbol{u}=\sum_{k=1}^{n}N_k(\xi,\eta,\zeta)\boldsymbol{u}_k \tag{8.2}$$

其中，$\boldsymbol{u}=\{u_1,u_2,u_3\}^T$ 为任意物质点的位移矢量，是 Lagrange 坐标的函数，$\boldsymbol{u}_k$ 是单元第 k 个节点对应物质点的位移。将（8.1）和（8.2）式统一写成矩阵形式，

$$\boldsymbol{X}=\boldsymbol{N}\boldsymbol{X}^e \quad \boldsymbol{u}=\boldsymbol{N}\cdot\boldsymbol{d}^e \tag{8.3}$$

式中，$\boldsymbol{X}^e=\{\boldsymbol{X}_1^T,\boldsymbol{X}_2^T,...,\boldsymbol{X}_n^T,\}^T$ 为单元节点坐标矢量，n 为单元节点数，$\boldsymbol{d}^e$ 为单元节点位移矢量。大变形情况下，Green 应变 $\boldsymbol{E}$ 可分解为线性和非线性两部分之和，

$$\boldsymbol{E}=\boldsymbol{E}_L+\boldsymbol{E}_N \tag{8.4}$$

式中，

$$\boldsymbol{E}=\{E_{11}\ \ E_{22}\ \ E_{33}\ \ 2E_{23}\ \ 2E_{31}\ \ 2E_{12}\}^T$$
$$\boldsymbol{E}_L=\{E_{11}^L\ \ E_{22}^L\ \ E_{33}^L\ \ 2E_{23}^L\ \ 2E_{31}^L\ \ 2E_{12}^L\}^T$$
$$\boldsymbol{E}_N=\{E_{11}^N\ \ E_{22}^N\ \ E_{33}^N\ \ 2E_{23}^N\ \ 2E_{31}^N\ \ 2E_{12}^N\}^T$$

根据 Green 应变和位移的关系式可以得出

$$E_{ij}^{L}=\frac{1}{2}\left(\frac{\partial u_i}{\partial X_j}+\frac{\partial u_j}{\partial X_i}\right) \tag{8.5}$$

$$E_{ij}^{N}=\frac{1}{2}\frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}=\frac{1}{2}\left(\frac{\partial \boldsymbol{u}}{\partial X_i}\right)^{T}\frac{\partial \boldsymbol{u}}{\partial X_j} \tag{8.6}$$

将（8.3）式代入（8.5）式和（8.6）式，可得

$$\boldsymbol{E}_L=\boldsymbol{B}_L\boldsymbol{d}^e \tag{8.7}$$

$$\boldsymbol{E}_N=\frac{1}{2}\boldsymbol{A}\cdot\boldsymbol{G}\cdot\boldsymbol{d}^e=\bar{\boldsymbol{B}}_N\boldsymbol{d}^e \tag{8.8}$$

式中，

$$\boldsymbol{B}_L=[\boldsymbol{B}_{L1},\ \boldsymbol{B}_{L2},\cdots,\boldsymbol{B}_{Ln}]$$

$$\boldsymbol{G}=[\boldsymbol{G}_1,\ \boldsymbol{G}_2,\ \cdots,\ \boldsymbol{G}_n]$$

$$\boldsymbol{A}^T=\begin{bmatrix}\dfrac{\partial \boldsymbol{u}}{\partial X_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dfrac{\partial \boldsymbol{u}}{\partial X_3} & \dfrac{\partial \boldsymbol{u}}{\partial X_2}\\ \boldsymbol{0} & \dfrac{\partial \boldsymbol{u}}{\partial X_2} & \boldsymbol{0} & \dfrac{\partial \boldsymbol{u}}{\partial X_3} & \boldsymbol{0} & \dfrac{\partial \boldsymbol{u}}{\partial X_1}\\ \boldsymbol{0} & \boldsymbol{0} & \dfrac{\partial \boldsymbol{u}}{\partial X_3} & \dfrac{\partial \boldsymbol{u}}{\partial X_2} & \dfrac{\partial \boldsymbol{u}}{\partial X_1} & \boldsymbol{0}\end{bmatrix}$$

$$\boldsymbol{B}_{Lk}=\begin{bmatrix}\dfrac{\partial N_k}{\partial X_1} & 0 & 0 & 0 & \dfrac{\partial N_k}{\partial X_3} & \dfrac{\partial N_k}{\partial X_2}\\ 0 & \dfrac{\partial N_k}{\partial X_2} & 0 & \dfrac{\partial N_k}{\partial X_3} & 0 & \dfrac{\partial N_k}{\partial X_1}\\ 0 & 0 & \dfrac{\partial N_k}{\partial X_3} & \dfrac{\partial N_k}{\partial X_2} & \dfrac{\partial N_k}{\partial X_1} & 0\end{bmatrix}^T$$

$$\boldsymbol{G}_k=\begin{bmatrix}\dfrac{\partial N_k}{\partial X_1}\boldsymbol{I}_{3\times3} & \dfrac{\partial N_k}{\partial X_2}\boldsymbol{I}_{3\times3} & \dfrac{\partial N_k}{\partial X_3}\boldsymbol{I}_{3\times3}\end{bmatrix}^T$$

矩阵 $\boldsymbol{B}_L$ 是 Green 应变线性部分与单元节点位移之间的转换矩阵。将（8.7）式和（8.8）式代入（8.4）式，可得

$$\boldsymbol{E}=(\boldsymbol{B}_L+\bar{\boldsymbol{B}}_N)\boldsymbol{d}^e=\bar{\boldsymbol{B}}\boldsymbol{d}^e \tag{8.9}$$

式中，

$$\bar{\boldsymbol{B}}=\boldsymbol{B}_L+\frac{1}{2}\boldsymbol{A}\cdot\boldsymbol{G} \tag{8.10}$$

式(8.9)给出 Green 应变和单元节点位移矢量之间的关系。由于 $\bar{\boldsymbol{B}}_N$ 中的 $\boldsymbol{A}$ 和位移有关，因而 Green 应变矢量和单元节点位移矢量之间是非线性关系。为便于应用虚功原理，将 Green 应变式(8.9)两边同时变分，可得

$$\delta\boldsymbol{E}=\boldsymbol{B}_L\cdot\delta\boldsymbol{d}^e+\bar{\boldsymbol{B}}_N\delta\boldsymbol{d}^e+\delta\bar{\boldsymbol{B}}_N\cdot\boldsymbol{d}^e$$

容易直接证明，

$$\bar{\boldsymbol{B}}_N \cdot \delta \boldsymbol{d}^e = \delta \bar{\boldsymbol{B}}_N \cdot \boldsymbol{d}^e$$

则

$$\delta \boldsymbol{E} = (\boldsymbol{B}_L + 2\bar{\boldsymbol{B}}_N)\delta \boldsymbol{d}^e = \boldsymbol{B} \cdot \delta \boldsymbol{d}^e \tag{8.11}$$

其中，

$$\boldsymbol{B} = \boldsymbol{B}_L + \boldsymbol{A} \cdot \boldsymbol{G} = \boldsymbol{B}_L + \boldsymbol{B}_N \tag{8.12}$$

将（8.11）式代入由 Green 应变和 Kirchhoff 应力表述的虚功方程，

$$\int_{V_0^e} (\delta \boldsymbol{E})^T \boldsymbol{S} \mathrm{d}V_0 = \delta \boldsymbol{d}^e \int_{V_0^e} \boldsymbol{B}^T \boldsymbol{S} \mathrm{d}V_0 = \delta \boldsymbol{d}^e \boldsymbol{F}^e \tag{8.13}$$

其中，$\boldsymbol{F}^e$ 为单元外部荷载的等效节点力，$S = \{S_{11}, S_{22}, S_{33}, S_{23}, S_{31}, S_{12}\}^T$ 为 Kirchhoff 应力矢量。为简洁起见，这里省略了单元间相互作用引起的等效节点力，以后将不再说明。由于 $\delta \boldsymbol{d}^e$ 的任意性，容易得到

$$\boldsymbol{\psi}(\boldsymbol{d}^e) = \int_{V_0^e} \boldsymbol{B}^T \cdot \boldsymbol{S} \mathrm{d}V_0 - \boldsymbol{F}^e = \boldsymbol{0} \tag{8.14}$$

组集后可得到离散系统的平衡方程，

$$\boldsymbol{\psi}(\boldsymbol{d}) = \int_{V_0} \boldsymbol{B}^T \cdot \boldsymbol{S} \mathrm{d}V_0 - \boldsymbol{F} = \boldsymbol{0} \tag{8.15}$$

式中，$\boldsymbol{F}$ 为有限元离散系统的外力等效节点力。上式即为有限元系统的位移形式的平衡方程组。

设材料的本构关系满足

$$\boldsymbol{S} = \boldsymbol{D} \cdot \boldsymbol{E} \tag{8.16}$$

代入（8.14）式可得

$$\boldsymbol{K}^e(\boldsymbol{d}^e) \cdot \boldsymbol{d}^e = \boldsymbol{F}^e \tag{8.17}$$

式中，

$$\boldsymbol{K}^e(\boldsymbol{d}^e) = \int_{V_0^e} \boldsymbol{B}^T \boldsymbol{D} \cdot \bar{\boldsymbol{B}} \mathrm{d}V_0 \tag{8.18}$$

为单元的割线刚度阵。注意到

（1）$\boldsymbol{B}$ 和 $\bar{\boldsymbol{B}}$ 为位移的函数，因此方程（8.17）为非线性方程组；

（2）大变形情况下，单元的割线刚度矩阵 $\boldsymbol{K}^e$ 是非对称矩阵。

为便于采用 Newton 法求解非线性方程组，我们需要继续推导单元的切线刚度矩阵 $\boldsymbol{K}_T^e$，对（8.14）求全微分，得

$$\mathrm{d}\boldsymbol{\psi}(\boldsymbol{d}^e) = \int_{V_0^e} \boldsymbol{B}^T \mathrm{d}\boldsymbol{S} \mathrm{d}V_0 + \int_{V_0^e} \mathrm{d}\boldsymbol{B}^T \boldsymbol{S} \mathrm{d}V_0 \tag{8.19}$$

由（8.16）式可得材料的增量型本构关系，

$$\mathrm{d}\boldsymbol{S} = \boldsymbol{D}_T \mathrm{d}\boldsymbol{E} \tag{8.20}$$

将上式代入(8.19)式右端第一项，得

$$\int_{V_0^e} \boldsymbol{B}^T \mathrm{d}\boldsymbol{S} \mathrm{d}V_0 = \int_{V_0^e} \boldsymbol{B}^T \boldsymbol{D}_T \mathrm{d}\boldsymbol{E} \mathrm{d}V_0 = \int_{V_0^e} \boldsymbol{B}^T \boldsymbol{D}_T \boldsymbol{B} \cdot \mathrm{d}\boldsymbol{d}^e \mathrm{d}V_0 = \boldsymbol{K}_D^e \mathrm{d}\boldsymbol{d}^e \tag{8.21}$$

其中，

$$\boldsymbol{K}_D^e = \int_{V_0^e} \boldsymbol{B}^T \boldsymbol{D}_T \boldsymbol{B} \mathrm{d}V_0 \tag{8.22}$$

式中，$\boldsymbol{K}_D^e$ 是与本构矩阵相关的切线刚度阵。将(8.12)式代入上式，可得

$$\boldsymbol{K}_D^e = \boldsymbol{K}_{DL}^e + \boldsymbol{K}_{DN}^e \tag{8.23}$$

其中，

$$\boldsymbol{K}_{DL}^e = \int_{V_0^e} \boldsymbol{B}_L^T \boldsymbol{D}_T \boldsymbol{B}_L \mathrm{d}V_0 \tag{8.24}$$

$$\boldsymbol{K}_{DN}^e = \int_{V_0^e} (\boldsymbol{B}_L^T \boldsymbol{D}_T \boldsymbol{B}_N + \boldsymbol{B}_N^T \boldsymbol{D}_T \boldsymbol{B}_L + \boldsymbol{B}_N^T \boldsymbol{D}_T \boldsymbol{B}_N) \mathrm{d}V_0 \tag{8.25}$$

显然，$\boldsymbol{K}_{DL}^e$ 就是通常的小位移刚度矩阵，$\boldsymbol{K}_{DN}^e$ 是由大位移引起的，通常称为初位移矩阵或大位移矩阵。

现在我们来分析(8.19)式右端的第二项，由于 $\boldsymbol{B}_L$ 和 $\boldsymbol{G}$ 与节点位移 $\boldsymbol{d}^e$ 无关，因此有

$$\mathrm{d}\boldsymbol{B}^T \boldsymbol{S} = (\mathrm{d}\boldsymbol{A} \cdot \boldsymbol{G})^T \cdot \boldsymbol{S} = \boldsymbol{G}^T \cdot (\mathrm{d}\boldsymbol{A})^T \cdot \boldsymbol{S}$$

容易直接验证，

$$\mathrm{d}\boldsymbol{A}^T \cdot \boldsymbol{S} = \hat{\boldsymbol{M}} \cdot \boldsymbol{G}\, \mathrm{d}\boldsymbol{d}^e$$

其中，

$$\hat{\boldsymbol{M}} = \begin{bmatrix} S_{11}\boldsymbol{I}_{3\times3} & S_{12}\boldsymbol{I}_{3\times3} & S_{13}\boldsymbol{I}_{3\times3} \\ S_{12}\boldsymbol{I}_{3\times3} & S_{22}\boldsymbol{I}_{3\times3} & S_{23}\boldsymbol{I}_{3\times3} \\ S_{13}\boldsymbol{I}_{3\times3} & S_{23}\boldsymbol{I}_{3\times3} & S_{33}\boldsymbol{I}_{3\times3} \end{bmatrix}$$

于是有

$$\int_{V_0^e} \mathrm{d}\boldsymbol{B}^T \boldsymbol{S} \mathrm{d}V_0 = \boldsymbol{K}_S^e\, \mathrm{d}\boldsymbol{d}^e \tag{8.26}$$

式中，

$$\boldsymbol{K}_S^e = \int_{V_0^e} \boldsymbol{G}^T \hat{\boldsymbol{M}} \boldsymbol{G}\, \mathrm{d}V_0 \tag{8.27}$$

矩阵 $\boldsymbol{K}_S^e$ 是由于应力状态 $\boldsymbol{S}$ 引起的切线刚度，通常称为几何矩阵或初应力矩阵。将(8.21)和(8.26)式代入(8.19)式可得

$$\mathrm{d}\boldsymbol{\psi}(\boldsymbol{d}^e) = \boldsymbol{K}_T^e \cdot \mathrm{d}\boldsymbol{d}^e \tag{8.28}$$

$$\boldsymbol{K}_T^e = \boldsymbol{K}_{DL}^e + \boldsymbol{K}_{DN}^e + \boldsymbol{K}_S^e \tag{8.29}$$

$\boldsymbol{K}_T^e$ 即为单元的切线刚度矩阵。注意到 $\hat{\boldsymbol{M}}$ 是对称矩阵，若 $\boldsymbol{D}_T$ 为对称矩阵，则 $\boldsymbol{K}_T^e$ 也为对称矩阵，这一点与割线刚度矩阵(8.18)式是不同的，也正是我们希望的。

注意到几何矩阵对单元的切线刚度矩阵有较大的影响，以简单拉压的杆件为例，压杆中压应力的存在可能会导致单元切线刚度矩阵奇异，从而出现压杆失稳。另外，在计算单元的切线刚度矩阵时，应适当提高数值积分的精度阶，以满足几何非线性对计算精度的要求。

单元切线刚度矩阵组集后可得离散系统的切线刚度阵 $\boldsymbol{K}_T$。Newton 法求解几何非线性问题中常采用如下格式，

$$\boldsymbol{K}_T^{(k)} \cdot \Delta\boldsymbol{d}^{(k)} = \boldsymbol{F} - \int_{V_0} \boldsymbol{B}^T \boldsymbol{S}^{(k)} \mathrm{d}V_0 \tag{8.30}$$

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} + \Delta\boldsymbol{d}^{(k)} \tag{8.31}$$

式中，$k = 0,1,2,3,\ldots$，上标(k)代表第 k 个迭代步对应的物理量，$\boldsymbol{K}_T^{(k)} = \boldsymbol{K}_T(\boldsymbol{d}^{(k)})$。实际分析中，通常取 $\boldsymbol{d}^{(0)} = \boldsymbol{0}$。$\int_{V_0} \boldsymbol{B}^T \boldsymbol{S}^{(k)} \mathrm{d}V_0$ 代表第 k 迭代步的内力等效节点力，因此，式(8.30)的右端代表残差，表征迭代过程中的不平衡量，可以作为迭代收敛的判据。

注意，本节的推导过程中采用了 Kirchhoff 应力和 Green 应变间的本构关系，但大部分实际工程问题一般无法直接给出这样的本构关系。因此，需要将工程中的应力应变关系适当改造，以满足分析要求。

8.2　大变形问题的增量解法——T.L.法

许多大变形问题同时又属材料非线性问题（弹塑性问题，粘弹塑性问题等），其变形往往与时间或变形历史有关，对于这些问题必须使用增量方法求解。将关心的时间段离散为某个时间序列，

$$0 = t_0 < t_1 < t_2 < \cdots < t_N = T$$

要注意的是，只有在与时间相关的问题（如非稳定渗流问题、徐变问题、瞬态响应问题）中，时间具有真正时间意义，其他情况下时间序列只是反应荷载变化过程的参数序列，可能与真实时间没有任何联系，此时时间序列只是对应一个真实的载荷序列。

现考虑时间序列（或载荷序列）的一个典型时刻 t_k，设从 $t = t_0$ 到 $t = t_k$ 的所有时刻的状态量均已知，现求解 $t_{k+1} = t_k + \Delta t_k$ 时刻的状态量。选取 $t = 0$ 时刻的构形，即初始构形，作为参考构形，该参考构形不随时间变化而变化。这种始终以不变化的初始构形作为参考构形的表述方法称为完全的拉格朗日（Total Lagrangian）方法，简记为 T.L.法。

$t = t_k$ 和 t_{k+1} 时刻的 Green 应变可以分别表示为

$$E_{ij}^{k} = \frac{1}{2}\left(\frac{\partial u_i^k}{\partial X_j} + \frac{\partial u_j^k}{\partial X_i} + \frac{\partial u_m^k}{\partial X_i}\frac{\partial u_m^k}{\partial X_j}\right) \tag{8.32}$$

$$E_{ij}^{k+1} = \frac{1}{2}\left(\frac{\partial u_i^{k+1}}{\partial X_j} + \frac{\partial u_j^{k+1}}{\partial X_i} + \frac{\partial u_m^{k+1}}{\partial X_i}\frac{\partial u_m^{k+1}}{\partial X_j}\right) \tag{8.33}$$

t_{k+1} 时刻的应变也可以用 t_k 时刻的应变及其增量之和的形式表示为

$$E_{ij}^{k+1} = E_{ij}^{k} + \Delta E_{ij}^{k} \tag{8.34}$$

同样位移也可以用增量形式表示为

$$u_i^{k+1} = u_i^k + \Delta u_i^k \tag{8.35}$$

容易导出应变增量的表达式为

$$\Delta E_{ij}^{k} = \left(\Delta E_{ij}^{k}\right)_{L_0} + \left(\Delta E_{ij}^{k}\right)_{L_1} + \left(\Delta E_{ij}^{k}\right)_{N} \tag{8.36}$$

式中，

$$\left(\Delta E_{ij}^{k}\right)_{L_0} = \frac{1}{2}\left(\frac{\partial \Delta u_i^k}{\partial X_j} + \frac{\partial \Delta u_j^k}{\partial X_i}\right) \tag{8.37}$$

$$\left(\Delta E_{ij}^{k}\right)_{L_1} = \frac{1}{2}\left(\frac{\partial u_m^k}{\partial X_i}\frac{\partial \Delta u_m^k}{\partial X_j} + \frac{\partial \Delta u_m^k}{\partial X_i}\frac{\partial u_m^k}{\partial X_j}\right) \tag{8.38}$$

$$\left(\Delta E_{ij}^{k}\right)_{N} = \frac{1}{2}\frac{\partial \Delta u_m^k}{\partial X_i}\frac{\partial \Delta u_m^k}{\partial X_j} \tag{8.39}$$

由于t_k时刻的位移$\boldsymbol{u}^k$是已知的，$\left(\Delta E_{ij}^k\right)_{L_0}$和$\left(\Delta E_{ij}^k\right)_{L_1}$与未知的位移增量$\Delta u_i^k$呈线性关系，而$\left(\Delta E_{ij}^k\right)_N$与$\Delta u_i^k$呈非线性关系，是增量应变的非线性部分。

将（8.36）式写成矩阵形式，

$$\Delta \boldsymbol{E}^k = \Delta \boldsymbol{E}_{L_0}^k + \Delta \boldsymbol{E}_{L_1}^k + \Delta \boldsymbol{E}_N^k \tag{8.40}$$

式中，

$$\Delta \boldsymbol{E}_{L_0}^k = \boldsymbol{L} \cdot \Delta \boldsymbol{u}^k \tag{8.41}$$

$$\Delta \boldsymbol{E}_{L_1}^k = \boldsymbol{A}^k \cdot \boldsymbol{H} \cdot \Delta \boldsymbol{u}^k \tag{8.42}$$

$$\Delta \boldsymbol{E}_N^k = \frac{1}{2} \Delta \boldsymbol{A}^k \cdot \boldsymbol{H} \cdot \Delta \boldsymbol{u}^k \tag{8.43}$$

其中，

$$\left(\Delta \boldsymbol{A}^k\right)^T = \begin{bmatrix} \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_1} & 0 & 0 & 0 & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_3} & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_2} \\ 0 & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_2} & 0 & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_3} & 0 & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_1} \\ 0 & 0 & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_3} & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_2} & \dfrac{\partial \Delta \boldsymbol{u}^k}{\partial X_1} & 0 \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_{3\times 3} \dfrac{\partial}{\partial X_1} & \boldsymbol{I}_{3\times 3} \dfrac{\partial}{\partial X_2} & \boldsymbol{I}_{3\times 3} \dfrac{\partial}{\partial X_3} \end{bmatrix}^T$$

由位移插值模式（8.3）可得位移增量在单元内的插值形式，

$$\Delta \boldsymbol{u}^k = \boldsymbol{N} \cdot (\Delta \boldsymbol{d}^k)^e \tag{8.44}$$

将（8.44）式代入（8.41）、（8.42）和（8.43）式，然后代入（8.40）式可得

$$\Delta \boldsymbol{E}^k = \bar{\boldsymbol{B}}^k (\Delta \boldsymbol{d}^k)^e \tag{8.45}$$

$$\delta(\Delta \boldsymbol{E}^k) = \boldsymbol{B}^k \, \delta(\Delta \boldsymbol{d}^k)^e \tag{8.46}$$

其中，

$$\bar{\boldsymbol{B}}^k = \boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k + \bar{\boldsymbol{B}}_N^k$$

$$\boldsymbol{B}^k = \boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k + \boldsymbol{B}_N^k$$

$$\boldsymbol{B}_{L_0} = \boldsymbol{L} \cdot \boldsymbol{N} \qquad \boldsymbol{B}_{L_1}^k = \boldsymbol{A}^k \cdot \boldsymbol{G} \qquad \boldsymbol{B}_N^k = 2\bar{\boldsymbol{B}}_N^k = \Delta \boldsymbol{A}^k \boldsymbol{G}$$

注意到$\boldsymbol{B}_{L_0}$是与位移位移增量无关的线性部分，其在形式上与小变形中的应变-位移转移矩阵$\boldsymbol{B}$相同；$\boldsymbol{B}_{L_1}^k$与位移增量无关，但与前一步的位移$\boldsymbol{u}^k$有关，表示在增量应变的线性部分中的初位移效应。

类似地，t_{k+1}时的 Kirchhoff 应力也可以表示为t_k时刻的应力与应力增量之和的形式，即

$$\boldsymbol{S}^{k+1} = \boldsymbol{S}^k + \Delta \boldsymbol{S}^k \tag{8.47}$$

注意到t_k时刻的位移和应变是已知的，因此，

$$\delta \boldsymbol{u}^{k+1} = \delta(\boldsymbol{u}^k + \Delta \boldsymbol{u}^k) = \delta(\Delta \boldsymbol{u}^k) \tag{8.48}$$

$$\delta \boldsymbol{E}^{k+1} = \delta(\boldsymbol{E}^k + \Delta \boldsymbol{E}^k) = \delta(\Delta \boldsymbol{E}^k) \tag{8.49}$$

代入 t_{k+1} 时刻的虚功方程的左端，可得

$$\int_{V_0^e} (\delta \boldsymbol{E}^{k+1})^T \boldsymbol{S}^{k+1} \mathrm{d}V_0 = \left(\delta(\Delta \boldsymbol{d}^k)^e\right)^T \int_{V_0^e} (\boldsymbol{B}^k)^T (\boldsymbol{S}^k + \Delta \boldsymbol{S}^k) \mathrm{d}V_0 \tag{8.50}$$

根据 $\delta(\Delta \boldsymbol{d}^e)^k$ 的任意性，可得单元体的平衡方程，经组集得离散系统的平衡方程，

$$\boldsymbol{\psi} = \int_{V_0} (\boldsymbol{B}^k)^T (\boldsymbol{S}^k + \Delta \boldsymbol{S}^k) \mathrm{d}V_0 - \boldsymbol{F}^{k+1} = \boldsymbol{0} \tag{8.51}$$

将 $\boldsymbol{B}^k$ 的表达式代入上式，

$$\boldsymbol{\psi} = \int_{V_0} (\boldsymbol{B}^k)^T \Delta \boldsymbol{S}^k \mathrm{d}V_0 + \int_{V_0} (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k)^T \boldsymbol{S}^k \mathrm{d}V_0 + \int_{V_0} (\boldsymbol{B}_N^k)^T \boldsymbol{S}^k \mathrm{d}V_0 - \boldsymbol{F}^{k+1} = \boldsymbol{0} \tag{8.52}$$

注意到

$$(\boldsymbol{B}_N^k)^T \boldsymbol{S}^k = (\Delta \boldsymbol{A}^k \cdot \boldsymbol{G})^T \cdot \boldsymbol{S}^k = \boldsymbol{G}^T (\Delta \boldsymbol{A}^k)^T \boldsymbol{S}^k = \boldsymbol{G}^T \cdot \hat{\boldsymbol{M}}^k \cdot \boldsymbol{G} \cdot \Delta \boldsymbol{d}^k \tag{8.53}$$

其中 $\hat{\boldsymbol{M}}^k$ 为 $\hat{\boldsymbol{M}}$ 在 t_k 时刻的值，由于 t_k 时刻的 Kirchhoff 应力为已知量，因此 $\hat{\boldsymbol{M}}^k$ 在计算过程中也是已知量。如果记

$$\boldsymbol{K}_s^k = \int_{V_0} \boldsymbol{G}^T \hat{\boldsymbol{M}}^k \boldsymbol{G} \mathrm{d}V_0 \tag{8.54}$$

$$\boldsymbol{F}_s^k = \int_{V_0} (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k)^T \boldsymbol{S}^k \mathrm{d}V_0 \tag{8.55}$$

则有

$$\boldsymbol{\psi}(\Delta \boldsymbol{d}^k) = \int_{V_0} (\boldsymbol{B}^k)^T \Delta \boldsymbol{S}^k \mathrm{d}V_0 + \boldsymbol{K}_s^k \Delta \boldsymbol{d}^k + \boldsymbol{F}_s^k - \boldsymbol{F}^{k+1} = \boldsymbol{0} \tag{8.56}$$

称 $\boldsymbol{K}_s^k$ 为初应力矩阵或几何矩阵，有时也称为非线性应变增量刚度矩阵，$\boldsymbol{F}_s$ 是 t_k 时刻的应力场对应的等效节点力矢量。

为方便求解非线性方程组（8.56）式，将积分式中的位移应变和应力应变增量关系线性化，即假设

$$\Delta \boldsymbol{E}^k = (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k + \boldsymbol{B}_N^k) \Delta \boldsymbol{d}^k \approx (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k) \Delta \boldsymbol{d}^k \tag{8.57}$$

$$\Delta \boldsymbol{S}^k \approx \boldsymbol{D}_T^k \cdot \Delta \boldsymbol{E}^k \approx \boldsymbol{D}_T^k (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}^k) \Delta \boldsymbol{d}^k \tag{8.58}$$

则（8.56）式可以简化为

$$(\boldsymbol{K}_L^k + \boldsymbol{K}_s^k) \Delta \boldsymbol{d}^k = \boldsymbol{F}^{k+1} - \boldsymbol{F}_s^k \tag{8.59}$$

其中，

$$\boldsymbol{K}_L^k = \int_{V_0} (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1})^T \boldsymbol{D}_T^k (\boldsymbol{B}_{L_0} + \boldsymbol{B}_{L_1}) \mathrm{d}V_0 \tag{8.60}$$

线性化假定可能会引起解的偏移，为避免产生太大的计算误差，通常需要在每一时间步或载荷增量步内采用 Newton 或修正的 Newton 方法迭代计算。下面给出的是修正 Newton 方法的迭代公式，

$$(\boldsymbol{K}_L^k + \boldsymbol{K}_s^k) \Delta \boldsymbol{d}^{k(m)} = \boldsymbol{F}^{k+1} - \boldsymbol{F}_s^{k+1(m)} \tag{8.61}$$

式中，

$$\boldsymbol{F}_S^{k+1(m)} = \int_{V_0} (\boldsymbol{B}^{k+1(m)})^T \boldsymbol{S}^{k+1(m)} \mathrm{d}V \tag{8.62}$$

$$\boldsymbol{d}^{k+1(m+1)} = \boldsymbol{d}^{k+1(m)} + \Delta \boldsymbol{d}^{k(m)} \tag{8.63}$$

$$\boldsymbol{d}^{k+1(0)} = \boldsymbol{d}^{k} \tag{8.64}$$

其中，$m = 0,1,2,\cdots$ 为迭代步。

8.3 大变形问题的增量解法——U.L.法

上节介绍大变形问题完全的拉格朗日增量解法，该方法始终以初始构形作为描述任意时刻变形的参考构形。其实，我们在求解 t_{k+1} 时刻的状态量时，从 t_0 到 t_k 的所有时刻的构形都是已知的，我们完全可以选用其中某一个时刻的构形作为描述 t_{k+1} 时刻物体变形的参考构形。一种常用的方法就是以 t_k 时刻的构形作为参考构形。我们称这一描述方法为修正的拉格朗日法（Updated Lagrangian 法），简称为 U.L.法。与 T.L.方法不同，U.L.法需要不断地修正参考构形。采用这一方法，$[t_k, t_k + \Delta t_k]$ 时间段内的格林应变是相对于 t_k 时刻的状态定义的。因此，

$$\Delta E_{ij}^{k} = \frac{1}{2}\left(\frac{\partial \Delta u_j^k}{\partial x_i^k} + \frac{\partial \Delta u_i^k}{\partial x_j^k} + \frac{\partial \Delta u_m^k}{\partial x_i^k}\frac{\partial \Delta u_m^k}{\partial x_j^k}\right) \tag{8.65}$$

类似地，将（8.65）式分解为线性和非线性两部分之和，

$$\Delta E_{ij}^{k} = \left(\Delta E_{ij}^{k}\right)_L + \left(\Delta E_{ij}^{k}\right)_L \tag{8.66}$$

其中，

$$\left(\Delta E_{ij}^{k}\right)_L = \frac{1}{2}\left(\frac{\partial \Delta u_i^k}{\partial x_j^k} + \frac{\partial \Delta u_j^k}{\partial x_i^k}\right) \tag{8.67}$$

$$\left(\Delta E_{ij}^{k}\right)_N = \frac{1}{2}\frac{\partial \Delta u_m^k}{\partial x_i^k}\frac{\partial \Delta u_m^k}{\partial x_j^k} \tag{8.68}$$

比较（8.67）式和（8.37）、（8.38）式可见，这里的应变增量线性部分要比 T.L.法中的线性部分简单。因为这里没有涉及初位移 u_i^k 的效应。将（8.66）、（8.67）和（8.68）式用矢量形式表示，比较上一节的推导容易得出

$$\Delta \boldsymbol{E}^{k} = \Delta \boldsymbol{E}_L^{k} + \Delta \boldsymbol{E}_N^{k} = \bar{\boldsymbol{B}}(\Delta \boldsymbol{d}^{e})^{k} \tag{8.69}$$

$$\Delta \boldsymbol{E}_L^{k} = \boldsymbol{B}_L(\Delta \boldsymbol{d}^{e})^{k} \tag{8.70}$$

$$\Delta \boldsymbol{E}_N^{k} = \bar{\boldsymbol{B}}_N \cdot (\Delta \boldsymbol{d}^{e})^{k} \tag{8.71}$$

其中，

$$\bar{\boldsymbol{B}} = \boldsymbol{B}_L + \bar{\boldsymbol{B}}_N \qquad \bar{\boldsymbol{B}}_N = \frac{1}{2}\Delta \boldsymbol{A}^{k} \cdot \boldsymbol{G}$$

对(8.70)和(8.71)式求变分，

$$\delta(\Delta \boldsymbol{E}_L^{k}) = \boldsymbol{B}_L \delta(\Delta \boldsymbol{d}^{k})^{e} \tag{8.72}$$

$$\delta(\Delta \boldsymbol{E}_N^{k}) = \boldsymbol{B}_N \delta(\Delta \boldsymbol{d}^{k})^{e} = \Delta \boldsymbol{A}^{k} \cdot \boldsymbol{G} \cdot \delta(\Delta \boldsymbol{d}^{k})^{e} \tag{8.73}$$

则应变增量的变分为

$$\delta(\Delta \boldsymbol{E}^{k}) = \boldsymbol{B}\delta(\Delta \boldsymbol{d}^{k})^{e} = (\boldsymbol{B}_L + \Delta \boldsymbol{A}^{k}\boldsymbol{G})\delta(\Delta \boldsymbol{d}^{k})^{e} \tag{8.74}$$

现在我们来分析 t_{k+1} 时刻以 t_k 时刻的构形为参考构形的虚功方程，

$$\int_{V^k}\delta(\boldsymbol{E}^{k+1})^T\boldsymbol{S}^{k+1}\mathrm{d}V=\int_{V^k}(\delta\boldsymbol{u}^{k+1})^T\boldsymbol{f}^{k+1}\mathrm{d}V+\int_{\Gamma_t^k}(\delta\boldsymbol{u}^{k+1})^T\cdot\boldsymbol{t}^{k+1}\cdot\mathrm{d}\Gamma \tag{8.75}$$

这里的 Kirchhoff 应力 $\boldsymbol{S}^{k+1}$，Green 应变 $\boldsymbol{E}^{k+1}$，荷载 $\boldsymbol{f}^{k+1}$ 和 $\boldsymbol{t}^{k+1}$ 均为 t_{k+1} 时刻的量，是以 t_k 时刻的构形为参考构形表述的。于是写成增量形式有

$$\boldsymbol{S}^{k+1}=\boldsymbol{S}^k+\Delta\boldsymbol{S}^k=\boldsymbol{\tau}^k+\Delta\boldsymbol{S}^k \tag{5.76}$$

$$\boldsymbol{E}^{k+1}=\boldsymbol{E}^k+\Delta\boldsymbol{E}^k=\boldsymbol{e}^k+\Delta\boldsymbol{E}^k \tag{8.77}$$

上式中 $\boldsymbol{\tau}^k$ 为 t_k 时刻的 Euler 应力矢量，$\boldsymbol{e}^k$ 为 t_k 时刻的 Almansi 应变矢量。注意 $\boldsymbol{e}^k$ 是已知的应变矢量，由（8.77）式容易得出

$$\delta\boldsymbol{E}^{k+1}=\delta(\Delta\boldsymbol{E}^k) \tag{8.78}$$

将（8.78）式代入（8.75）式，并注意到 $\delta(\Delta\mathbf{d}^k)^e$ 的任意性，可得单元的平衡方程，组集后可得有限元离散系统的平衡方程，

$$\int_{V^k}\boldsymbol{B}^T(\boldsymbol{\tau}^k+\Delta\boldsymbol{S}^k)\mathrm{d}V=\boldsymbol{F}^{k+1} \tag{8.79}$$

式中，$\boldsymbol{F}^{k+1}$ 为 t_{k+1} 时刻相对 t_k 时刻的构形定义的外力等效节点力矢量。将上式改写成如下形式，

$$\boldsymbol{\psi}(\Delta\boldsymbol{d}^k)=\int_{V^k}\boldsymbol{B}^T(\boldsymbol{\tau}^k+\Delta\boldsymbol{S}^k)\mathrm{d}V-\boldsymbol{F}^{k+1}=\boldsymbol{0} \tag{8.80}$$

为求解上述方程，需要引入应力应变关系。从上式可以看出，采用相对 t_k 时刻的构形定义的格林应变增量和克希荷夫应力增量（或以格林应变率和克希荷夫应力率）表述的本构方程是最合适的。遗憾的是，这样的本构关系很难直接试验给出。本构关系理论指出，大变形条件下的应力应变关系常用焦曼应力率和变形率张量表述，即

$$\overset{\nabla}{\tau}_{lm}=D^{(T)}_{lmpq}V_{pq} \tag{8.81}$$

写成矩阵形式为

$$\overset{\nabla}{\boldsymbol{\tau}}=\boldsymbol{D}_T\boldsymbol{V} \tag{8.82}$$

注意，上述本构形式不只适用于线弹性材料，也可应用于非线性弹性和弹塑性材料，对于弹塑性材料，$\boldsymbol{D}_T=\boldsymbol{D}_{ep}$。根据式(3.63)并注意到这里的格林应变是相对 t_k 时刻的构形定义的，于是在 t_k 时刻有

$$\boldsymbol{V}^k_{ij}=\dot{\boldsymbol{E}}^k_{ij} \tag{8.83}$$

类似地，利用式(4.64) 可导出在 t_k 时刻有

$$\dot{S}^k_{lm}=\overset{\nabla}{\tau}{}^k_{lm}-\tau^k_{lp}V^k_{pm}-\tau^k_{mp}V^k_{pl}+\tau^k_{lm}\frac{\partial v^k_p}{\partial x_p}=\overset{\nabla}{\tau}{}^k_{lm}-S^k_{lp}V^k_{pm}-S^k_{mp}V^k_{pl}+S^k_{lm}\frac{\partial v^k_p}{\partial x_p} \tag{8.84}$$

注意

$$\frac{\partial v_m}{\partial x_m}=V_{pp}=\delta_{pq}V_{pq} \tag{8.85}$$

并将(8.83)式代入(8.84)式，得

$$\overset{\nabla}{\tau}{}^k_{lm}=\dot{S}^k_{lm}+(S^k_{lp}\delta_{mq}+S^k_{mp}\delta_{lq}-S^k_{lm}\delta_{pq})\dot{E}_{pq} \tag{8.86}$$

再将(8.83)式和(8.86)式代入(8.81)式，得 t_k 时刻的本构关系，

$$\dot{S}_{lm}^{k} = \bar{D}_{lmpq}^{k}\dot{E}_{pq}^{k} \tag{8.87}$$

其中，

$$\bar{D}_{lmpq}^{k} = D_{lmpq}^{(T)} - S_{lp}^{k}\delta_{mq} - S_{mp}^{k}\delta_{lq} + S_{lm}^{k}\delta_{pq} \tag{8.88}$$

注意上式等号右端的最后一项使得本构张量 $\bar{D}_{lmpq}^{k}$ 非对称。对于金属材料的弹塑性问题，当弹性应变相对塑性应变很小时，可以假定变形是不可压缩的，此时可将非对称项略去，得到

$$\bar{D}_{lmpq}^{k} = D_{lmpq}^{(T)} - S_{lp}^{k}\delta_{mq} - S_{mp}^{k}\delta_{lq} \tag{8.89}$$

上面关于本构关系的推导在 t_k 时刻是准确成立的，在 t_k 的一个很小邻域内可以认为是近似成立的。为便于求解，将(8.80)式中的有限应力应变增量关系线性化，即

$$\Delta S_{lm}^{k} = (D_{lmpq}^{(T)} - S_{lp}^{k}\delta_{mq} - S_{mp}^{k}\delta_{lq})\Delta E_{pq}^{k} \tag{8.90}$$

将(8.90)式写成矩阵形式，

$$\Delta \boldsymbol{S}^{k} = (\boldsymbol{D}_{T} - \tilde{\boldsymbol{M}}^{k})\Delta \boldsymbol{E}^{k} \tag{8.91}$$

式中，

$$\tilde{\boldsymbol{M}}^{k} = \begin{bmatrix} 2\tau_{11}^{k} & 0 & 0 & 0 & \tau_{31}^{k} & \tau_{12}^{k} \\ 0 & 2\tau_{22}^{k} & 0 & \tau_{23}^{k} & 0 & \tau_{12}^{k} \\ 0 & 0 & 2\tau_{33}^{k} & \tau_{23}^{k} & \tau_{31}^{k} & 0 \\ 0 & \tau_{23}^{k} & \tau_{23}^{k} & \dfrac{\tau_{22}^{k}+\tau_{33}^{k}}{2} & \dfrac{\tau_{12}^{k}}{2} & \dfrac{\tau_{31}^{k}}{2} \\ \tau_{31}^{k} & 0 & \tau_{31}^{k} & \dfrac{\tau_{12}^{k}}{2} & \dfrac{\tau_{11}^{k}+\tau_{33}^{k}}{2} & \dfrac{\tau_{23}^{k}}{2} \\ \tau_{12}^{k} & \tau_{12}^{k} & 0 & \dfrac{\tau_{31}^{k}}{2} & \dfrac{\tau_{23}^{k}}{2} & \dfrac{\tau_{11}^{k}+\tau_{22}^{k}}{2} \end{bmatrix} \tag{8.92}$$

现在再来分析(8.80)式，将积分式中的积分核展开后，积分式中的第一项为

$$\int_{V^k} \boldsymbol{B}^{T}\boldsymbol{\tau}^{k}\mathrm{d}V = \int_{V^k} \boldsymbol{B}_{L}^{T}\boldsymbol{\tau}^{k}\mathrm{d}V + \int_{V^k} \boldsymbol{G}^{T}(\Delta \boldsymbol{A}^{k})^{T}\boldsymbol{\tau}^{k}\mathrm{d}V \tag{8.93}$$

$$\int_{V^k} \boldsymbol{G}^{T}(\Delta \boldsymbol{A}^{k})^{T}\boldsymbol{\tau}^{k}\mathrm{d}V = \int_{V^k} \boldsymbol{G}^{T}\hat{\boldsymbol{M}}^{k}\boldsymbol{G}\mathrm{d}V\Delta \boldsymbol{d}^{k} \tag{8.94}$$

将(8.91)式代入(8.80)式中积分式的第二项。为便于求解，进一步简化有限位移和有限应变间的关系，

$$\Delta \boldsymbol{E}^{k} = \boldsymbol{B}^{k}\Delta \boldsymbol{d}^{k} \tag{8.95}$$

于是(8.80)式可以简化为

$$(\boldsymbol{K}_{s} + \boldsymbol{K}_{LN})\Delta \boldsymbol{d}^{k} = \boldsymbol{F}^{k+1} - \boldsymbol{F}_{S}^{k} \tag{8.96}$$

式中，

$$\boldsymbol{K}_{s} = \int_{V^k} \boldsymbol{G}^{T}\hat{\boldsymbol{M}}^{k}\boldsymbol{G}\mathrm{d}V \tag{8.97}$$

$$\boldsymbol{K}_{LN} = \int_{V^k} \boldsymbol{B}^{T}(\boldsymbol{D}_{T} - \tilde{\boldsymbol{M}}^{k})\boldsymbol{B}\mathrm{d}V \tag{8.98}$$

$$\boldsymbol{F}_{S}^{k+1} = \int_{V^k} \boldsymbol{B}_{L}^{T}\boldsymbol{\tau}^{k}\mathrm{d}V \tag{8.99}$$

(8.98)式中的 $\boldsymbol{B}$ 可以进一步用 $\boldsymbol{B}_{L}$ 代替。

注意，在推导式(8.96)过程中作了一些线性化假定，因此，仅在较小的载荷增量条件下才不至于引起解的较大漂移。

在大变形增量问题的有限元分析中，使用 T.L.法还是 U.L.法主要取决于两个方面：第一个方面就是单元分析的计算效率。从两种方法的有限元列式可以看出，U.L.法的计算量比 T.L.法少，计算效率高。由于单元分析在整个有限元分析过程中占用的计算机时间相对较少，且随着计算机硬件的发展，单元分析的计算效率已不是考虑的重点。第二个方面就是材料的本构关系。如果材料的应力应变关系是用相对初始构形的克希荷夫应力定义的，T.L.法是很方便的；如果本构关系是用相对现时构形的欧拉应力给出的，最好采用 U.L.法。

8.4　2 节点大变形杆单元

本节介绍大变形条件下 2 节点杆单元的切线刚度矩阵（Lagrange 法）。

如图 8.1 所示的 2 节点杆单元，以节点 1 为原点、单元轴向为 X_1' 轴建立直角坐标系 $OX_1'X_2'X_3'$。记沿 X_1'、X_2' 和 X_3' 轴的位移为 u'、v' 和 w'。用矢量形式表示为

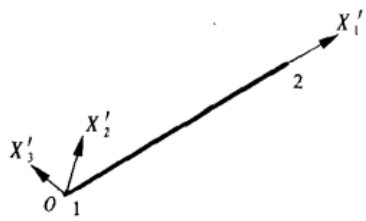


图8.1 2节点杆单元

$$\boldsymbol{u}' = \{u', v', w'\}^T \tag{8.100}$$

单元节点位移矢量可表示为

$$\boldsymbol{d}'^{e} = \{u_1', v_1', w_1', u_2', v_2', w_2'\}^T \tag{8.101}$$

单元内任意点的位移按下式插值得到

$$\boldsymbol{u}' = \boldsymbol{N}\boldsymbol{d}'^{e} \tag{8.102}$$

$$\boldsymbol{N} = [N_1\boldsymbol{I}_{3\times3}, N_2\boldsymbol{I}_{3\times3}] \tag{8.103}$$

其中，

$$N_1 = 1 - X_1'/L \tag{8.104}$$

$$N_2 = X_1'/L \tag{8.105}$$

式中，L 为初始构形中单元的长度。单元内的轴向 Green 应变为

$$E_{11} = \frac{\partial u_1'}{\partial X_1'} + \frac{1}{2}[(\frac{\partial u_1'}{\partial X_1'})^2 + (\frac{\partial u_2'}{\partial X_1'})^2 + (\frac{\partial u_3'}{\partial X_1'})^2] \tag{8.106}$$

写成矩阵形式为

$$E_{11} = \boldsymbol{L}\ \boldsymbol{u}' + \frac{1}{2}\boldsymbol{A}\ \boldsymbol{H}\ \boldsymbol{u}' \tag{8.107}$$

式中，

$$\boldsymbol{L} = [\frac{\partial}{\partial X_1'},\ 0,\ 0] \tag{8.108}$$

$$\boldsymbol{A} = [\frac{\partial u'}{\partial X_1'},\ \frac{\partial v'}{\partial X_1'},\ \frac{\partial w'}{\partial X_1'}] \tag{8.109}$$

$$\boldsymbol{H} = \boldsymbol{I}_{3\times3}\frac{\partial}{\partial X_1'} \tag{8.110}$$

将(8.102)、(8.103)式代入(8.107)式，可得

$$E_{11}=(\boldsymbol{B}_L+\frac{1}{2}\boldsymbol{AG})\ \boldsymbol{d}'^e=\bar{\boldsymbol{B}}\boldsymbol{d}'^e \tag{8.111}$$

式中，

$$\boldsymbol{B}_L=\boldsymbol{L}\ \boldsymbol{N}=\frac{1}{L}[-1,\ 0,\ 0,\ 1,\ 0,\ 0] \tag{8.112}$$

$$\boldsymbol{A}=[\varepsilon,\ \theta_z,\ -\theta_y] \tag{8.113}$$

$$\boldsymbol{G}=\frac{1}{L}[-\boldsymbol{I}_{3\times3},\ \boldsymbol{I}_{3\times3}] \tag{8.114}$$

$$\bar{\boldsymbol{B}}=\boldsymbol{B}_L+\frac{1}{2}\boldsymbol{AG} \tag{8.115}$$

其中，

$$\varepsilon=\frac{u_2'-u_1'}{L}\qquad \theta_z=\frac{v_2'-v_1'}{L}\qquad -\theta_y=\frac{w_2'-w_1'}{L} \tag{8.116}$$

将 Green 应变求变分，类似前面各节的推导可得

$$\delta E_{11}=(\boldsymbol{B}_L+\boldsymbol{AG})\delta\boldsymbol{d}'^e=\boldsymbol{B}\,\delta\boldsymbol{d}'^e \tag{8.117}$$

$$\boldsymbol{B}=\boldsymbol{B}_L+\boldsymbol{AG} \tag{8.118}$$

设单元的轴向 Kirchhoff 应力 S_{11} 和 Green 应变间满足如下关系，

$$S_{11}=E_S^0\ E_{11} \tag{8.119}$$

记初始构形中单元的截面面积为 A，将 (8.111)、(8.117)和(8.119)式代入虚功方程的左端可推得单元刚度矩阵，

$$\boldsymbol{K}'^e=E_S^0AL\boldsymbol{B}^T\bar{\boldsymbol{B}} \tag{8.120}$$

同样，单元的刚度矩阵是非对称的。

单元的切线刚度矩阵为

$$\boldsymbol{K}_T'^e=E_T^0AL\boldsymbol{B}^T\boldsymbol{B}+N\,L\,\boldsymbol{G}^T\boldsymbol{G} \tag{8.121}$$

式中，

$$N=S_{11}A \tag{8.122}$$

$$E_T^0=\frac{\mathrm{d}S_{11}}{\mathrm{d}E_{11}} \tag{8.123}$$

展开(8.121)式

$$\boldsymbol{K}_T'^e=\frac{EA}{L}\boldsymbol{K}_L'+\frac{N}{L}\boldsymbol{K}_S'+\frac{EA}{L}\boldsymbol{K}_N' \tag{8.124}$$

其中，

$$\boldsymbol{K}_L'=\begin{bmatrix}1&0&0&-1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\-1&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}\qquad \boldsymbol{K}_S'=\begin{bmatrix}1&0&0&-1&0&0\\0&1&0&0&-1&0\\0&0&1&0&0&-1\\-1&0&0&1&0&0\\0&-1&0&0&1&0\\0&0&-1&0&0&1\end{bmatrix}$$

$$\boldsymbol{K}'_N=\begin{bmatrix} 2\varepsilon+\varepsilon^2 & (1+\varepsilon)\theta_z & -(1+\varepsilon)\theta_y & -(2\varepsilon+\varepsilon^2) & -(1+\varepsilon)\theta_z & (1+\varepsilon)\theta_y \\ & \theta_z\theta_z & -\theta_y\theta_z & -(1+\varepsilon)\theta_z & -\theta_z\theta_z & \theta_y\theta_z \\ & & \theta_y\theta_y & (1+\varepsilon)\theta_y & \theta_y\theta_z & -\theta_y\theta_y \\ & & & 2\varepsilon+\varepsilon^2 & (1+\varepsilon)\theta_z & -(1+\varepsilon)\theta_y \\ & \text{对称} & & & \theta_z\theta_z & -\theta_y\theta_z \\ & & & & & \theta_y\theta_y \end{bmatrix}$$

在实际工程中，根据轴向变形 ε 和转动 θ_y 及 θ_z 间的相对量级关系，式(8.124)中的 $\boldsymbol{K}'_N$ 可进一步简化。

以上在单元局部坐标系中推导出杆单元的割线和切线刚度矩阵，它们不能直接参与组集和求解，需要经坐标变换，得到整体坐标系下的单元刚度矩阵。单元刚度阵的坐标变换方法和线性杆单元相同，读者可以参阅相关教材。

杆件的拉伸模量 E_S^0 和 E_T^0 一般通过材料的拉伸试验确定，而材料拉伸试验中采用的应力通常为 Lagrange 应力 Σ_{11}，应变为定义在初始构形上的 Cauchy 应变 $\vartheta=(l-L)/L$。其中，L 和 l 分别为试件变形前和变形后的长度。两者之间满足

$$\Sigma_{11}=E\vartheta$$

由此容易导出 E_S^0 和 E 间的关系式，

$$E_S^0=\frac{\sqrt{1+2E_{11}}-1}{E_{11}\sqrt{1+2E_{11}}}E=\frac{2}{(\lambda+1)\lambda}E \tag{8.125}$$

其中，

$$\lambda=\frac{l}{L}$$

当 E 为常数时，

$$E_T^0=(1+2E_{11})^{-\frac{3}{2}}E=\frac{E}{\lambda^3} \tag{8.126}$$

所附光盘中的文件 Elmt001.f90 就是 2 节点大变形杆单元的源代码。其中的主要模块说明如下。

```
      SUBROUTINE Elmt001( ielem, imats, iswth, iuses )
!.....
!       模块功能
!           2 节点杆单元，包括一维，二维或三维杆单元，可考虑大变形几何非线性，材
!       料为线性弹性。
!.....
      USE CtrlData
      USE ElmtData
      USE GlobData
      USE MeshData
      USE FactorData
```

```
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION direc( 3, 3 ), bmatx( 6 )

      IF( iswth .EQ. 0 ) THEN
!..... 读入材料信息并存储
        nprnc = ielem
        IF( nstrh .LT. 1 ) nstrh = 1
        IF( nprnc .NE. 0 ) WRITE( 12, 2000 )
        READ( 11, 2200 ) elast, varea, denst, itype, nlinc,                &
                         xbody, ybody, zbody, nbody
        IF( nprnc .NE. 0 )                                                 &
        WRITE( 12, 2400 ) elast, varea, denst, itype, nlinc,               &
                          xbody, ybody, zbody, nbody
        props(  2, imats ) = elast                    ! 拉伸模量
        props(  3, imats ) = varea                    ! 截面面积
        props(  4, imats ) = denst                    ! 质量密度
        props(  5, imats ) = itype                    ! 是否阻尼边界
        props(  6, imats ) = nlinc                    ! 是否考虑大变形
        props(  7, imats ) = xbody                    ! 体积力分量
        props(  8, imats ) = ybody                    ! 体积力分量
        props(  9, imats ) = zbody                    ! 体积力分量
        props( 10, imats ) = nbody                    ! 体积力载荷因子模式
        IF( nlinc .NE. 0 ) THEN
          IF( nincc .EQ. 3 ) nerrc = 45864
          nincc = 2
        END IF
        RETURN
      ELSE IF( iswth .EQ. 8 ) THEN
!..... 单元基本信息
        nnode = 2
        ndofn = ndime
        RETURN
      ELSE IF( iswth .EQ. 9 ) THEN
!..... 输出单元后处理网格
        IF( iuses .EQ. 0 ) RETURN
        WRITE( 15, 2600 ) 2, ielem, imats, 2, lnode( 1 ), lnode( 2 )
        RETURN
      ELSE IF( iswth .EQ. 21 ) THEN
!..... 辅助网格信息（适用改进等参元）
        WRITE( 25 ) 0
        RETURN
      END IF
!..... 其他功能选项从此开始
```

```
!..... 单元矩阵和矢量初始化
      DO idofe = 1, ndofe
        force( idofe ) = 0.0D0
        DO jdofe = 1, ndofe
          stife( idofe, jdofe ) = 0.0D0
        END DO
      END DO
      IF( iuses .EQ. 0 ) RETURN
!.....提取单元的基本信息
      elast = props( 2, imats )
      varea = props( 3, imats )
      denst = props( 4, imats )
      itype = props( 5, imats ) + 0.5D0
      nlinc = props( 6, imats ) + 0.5D0
      if( itype .GT. 0 ) itype = 1
      if( itype .LT. 0 ) itype =-1
!..... 确定单元的局部坐标、Kirchhoff 应力和 Green 应变间模量、单元应力
      elass = elast
      CALL Axis001( direc, fleng )
      CALL UpdateElast( elass, nlinc, 1 )        ! 割线模量
      CALL UpdateElast( elast, nlinc, 2 )        ! 切线模量
      CALL Strs001( direc, fleng, elass, vstrs, nlinc )
!..... 从此开始计算单元的元素矩阵
      IF( iswth .EQ. 1 .OR. iswth .EQ. 5 .OR. iswth .EQ. 10 ) THEN
!..... 单元切向刚度阵（iswth=1）、内力等效节点力（iswth=5）、节点反力（iswth=10）
!..... 先计算 ∫_{V0} B^T D B dV
        const = elast * varea / fleng
        CALL BMat001( bmatx, direc, nlinc, fleng )
        DO idofe = 1, ndime * 2
          DO jdofe = 1, ndime * 2
            stife( idofe, jdofe ) = bmatx( idofe ) *                    &
                                    const * bmatx( jdofe )
          END DO
        END DO
!..... 对大变形问题，再计算 ∫_{V0} G^T M G dV
        IF( nlinc .NE. 0 ) THEN
          const = vstrs * varea / fleng
          DO idime = 1, ndime
            idofe = idime + ndime
            stife( idime, idime ) = stife( idime, idime ) + const
            stife( idime, idofe ) = stife( idime, idofe ) - const
            stife( idofe, idime ) = stife( idofe, idime ) - const
```

```
            stife( idofe, idofe ) = stife( idofe, idofe ) + const
          END DO
        END IF
!..... 将单元刚度阵从局部坐标转换到整体坐标
        CALL Trans001( direc, 1, 0 )
!..... 计算内力等效节点力
        IF( nincc .EQ. 2 .OR. iswth .EQ. 10 ) THEN
          const = varea * vstrs
          DO idofe = 1, 2 * ndime
            force( idofe ) = force( idofe ) - bmatx( idofe ) * const
          END DO
          CALL Trans001( direc, 0, 1 )
!..... 将内力等效节点力从局部坐标转换到整体坐标
          IF( DABS( strsh( 1, ielem ) ) .GT. 1.0D-8 ) THEN
            vleng = 0.0D0
            DO idime = 1, ndime
              bmatx( idime ) = coren( idime, 2 ) - coren( idime, 1 ) +          &
                               dispe( ndofn + idime ) - dispe( idime )
              vleng = vleng + bmatx( idime ) * bmatx( idime )
            END DO
            vleng = DSQRT( vleng )
            DO idime = 1, ndime
              bmatx( idime ) = bmatx( idime ) / vleng
            END DO
            DO idime = 1, ndime
              idofe = ndofn + idime
              const = bmatx( idime ) * strsh( 1, ielem ) * varea
              force( idime ) = force( idime ) + const
              force( idofe ) = force( idofe ) - const
            END DO
          END IF
        END IF
      ELSE IF( iswth .EQ. 2 ) THEN
!..... 单元质量矩阵
        vmass = fleng * varea * denst / 2.0D0
        DO idofe = 1, ndofe
          stife( idofe, idofe ) = vmass
        END DO
      ELSE IF( iswth .EQ. 3 ) THEN
!..... 单元阻尼矩阵
        :
      ELSE IF( iswth .EQ. 4 ) THEN
!..... 单元几何刚度阵
        faxis = vstrs * varea / fleng
```

```
            DO idime = 1, ndime
              DO jdime = 1, ndime
                const = direc( idime, 1 ) * direc( jdime, 1 )
                stife( idime, jdime ) =-const * faxis
              END DO
              stife( idime, idime ) = stife( idime, idime ) + faxis
            END DO
            DO idime = 1, ndime
              DO jdime = 1, ndime
                iloca = idime + ndime
                jloca = jdime + ndime
                stife( iloca, jdime ) =-stife( idime, jdime )
                stife( idime, jloca ) =-stife( idime, jdime )
                stife( iloca, jloca ) = stife( idime, jdime )
              END DO
            END DO
          ELSE IF( iswth .EQ. 6 ) THEN
!..... 单元体积力等效节点
            nbody = props( 10, imats ) + 0.5D0
            CALL Factor( nbody )
            IF( nerrc .GT. 0 ) RETURN
            dvolu = varea * fleng * fctor / 2.0D0
            idofe = 0
            DO inode = 1, nnode
              DO idime = 1, ndime
                idofe = idofe + 1
                force( idofe ) = dvolu * props( idime + 6, imats )
              END DO
            END DO
          ELSE IF( MOD( iswth, mswth ) .EQ. 7 ) THEN
!..... 单元应力和内力
            faxis = vstrs * varea
            WRITE( 12, 2800 ) ielem, vstrs, faxis
          END IF
2000      FORMAT( 2x, '单元类型:                    杆单元' )
2200      FORMAT( 3F15.5, 2I5 / 3F15.5, I5 )
2400      FORMAT( 2x, '杨氏模量: ', E15.5/2x, '截面面积: ', E15.5 /          &
                  2x, '质量密度: ', E15.5/2x, '边界约束: ', I15    /          &
                  2x, '几何线性: ', I15   /2x, 'x-体积力: ', F15.5 /          &
                  2x, 'y-体积力: ', F15.5/2x, 'z-体积力: ', F15.5 /           &
                  2x, '动载类型: ', I15   )
2600      FORMAT( I3, 1X, I5, 1X, I4, 1X, I3, 8( 1X, I5 ) )
2800      FORMAT( 2x, '单元', I5, '   应力', E15.6, ' 轴力', E15.6 )
          RETURN
```

```
      END

      SUBROUTINE Axis001( direc, eleng )
!.....
!       模块功能
!           计算杆单元的局部坐标系 OX1'X2'X3' 和单元长度。
!.....
      USE CtrlData
      USE ElmtData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION direc( 3, 3 )
!.....初始化为 0
      DO idime = 1, 3
        DO jdime = 1, 3
          direc( idime, jdime ) = 0.0D0
        END DO
      END DO
!.....计算单元初始长度和轴线方向
      eleng = 0.0D0
      DO idime = 1, ndime
        direc( idime, 1 ) = coren( idime, 2 ) - coren( idime, 1 )
        eleng = eleng       + direc( idime, 1 ) * direc( idime, 1 )
      END DO
      eleng = DSQRT( eleng )
      DO idime = 1, ndime
        direc( idime, 1 ) = direc( idime, 1 ) / eleng
      END DO
!.....计算杆件在 X1X2 平面中的投影
      fleng = DSQRT( direc( 1, 1 ) * direc( 1, 1 ) +                    &
                     direc( 2, 1 ) * direc( 2, 1 ) )
      IF( fleng .LT. 1.0D-8 ) THEN
!.....如果投影长度为零，说明沿 X3 轴方向
        direc( 1, 2 ) = direc( 3, 1 )
        direc( 2, 3 ) = 1.0D0
      ELSE
!.....如果投影长度大于零
        direc( 1, 2 ) =-direc( 2, 1 ) / fleng
        direc( 2, 2 ) = direc( 1, 1 ) / fleng
        direc( 1, 3 ) =-direc( 2, 2 ) * direc( 3, 1 )
        direc( 2, 3 ) = direc( 1, 2 ) * direc( 3, 1 )
        direc( 3, 3 ) = direc( 1, 1 ) * direc( 2, 2 ) -                 &
                        direc( 1, 2 ) * direc( 2, 1 )
        fleng = DSQRT( direc( 1, 3 ) * direc( 1, 3 ) +                  &
```

```
                        direc( 2, 3 ) * direc( 2, 3 ) +                    &
                        direc( 3, 3 ) * direc( 3, 3 ) )
      direc( 1, 3 ) = direc( 1, 3 ) / fleng
      direc( 2, 3 ) = direc( 2, 3 ) / fleng
      direc( 3, 3 ) = direc( 3, 3 ) / fleng
    END IF
    RETURN
    END

    SUBROUTINE BMat001( bmatx, direc, nlinc, fleng )
!.....
!       模块功能
!           计算杆单元的 B_L （nlinc=0）或 B （nlinc≠0）
!.....
    USE CtrlData
    USE ElmtData
    IMPLICIT DOUBLE PRECISION( a-h, o-z )
    DIMENSION bmatx( 6 ), direc( 3, 3 ), displ( 3, 2 )
!..... 计算 B_L
    DO idofe = 1, 6
      bmatx( idofe ) = 0.0D0
    END DO
    bmatx(          1 ) = -1.0D0
    bmatx( ndime + 1 ) =  1.0D0
    IF( nlinc .EQ. 0 ) RETURN
!..... 计算 B_N
    DO inode = 1, 2
      DO idime = 1, ndime
        displ( idime, inode ) = 0.0D0
        DO jdime = 1, ndime
          idofe = ( inode - 1 ) * ndime + jdime
          displ( idime, inode ) = displ( idime, inode ) +                  &
          direc( jdime, idime ) * dispe( idofe )
        END DO
      END DO
    END DO
    DO idime = 1, ndime
      idofe = idime + ndime
      const = displ( idime, 2 ) - displ( idime, 1 )
      bmatx( idime ) = bmatx( idime ) - const / fleng
      bmatx( idofe ) = bmatx( idofe ) + const / fleng
    END DO
    RETURN
```

```
      END

      SUBROUTINE Strs001( direc, fleng, elast, vstrs, nlinc )
!.....
!       模块功能
!           计算杆单元的应力
!.....
      USE CtrlData
      USE ElmtData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION direc( 3, 3 ), displ( 3, 2 )
!..... 计算单元局部坐标系中的位移
      DO inode = 1, 2
        DO idime = 1, ndime
          displ( idime, inode ) = 0.0D0
          DO jdime = 1, ndime
            idofe = ( inode - 1 ) * ndime + jdime
            displ( idime, inode ) = displ( idime, inode ) +                    &
            direc( jdime, idime ) * dispe( idofe )
          END DO
        END DO
      END DO
!..... 计算ε、θz和-θy
      DO idime = 1, ndime
        displ( idime, 1 ) = displ( idime, 2 ) - displ( idime, 1 )
        displ( idime, 1 ) = displ( idime, 1 ) / fleng
      END DO
!..... 计算单元应变，先计算线性应变
      vstrn = displ( 1, 1 )
      IF( nlinc .NE. 0 ) THEN
!.....如果考虑几何非线性，再计算 Green 应变的非线性部分
        DO idime = 1, ndime
          vstrn = vstrn + displ( idime, 1 ) *                                  &
                  0.5D0 * displ( idime, 1 )
        END DO
      END IF
!.....计算 Kirchhoff 应力
      vstrs = vstrn * elast
      RETURN
      END

      SUBROUTINE Trans001( direc, nstif, nforc )
!.....
!       模块功能
```

```
!              单元矩阵和矢量的坐标变换
!.....
        USE CtrlData
        USE ElmtData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION direc( 3, 3 ), vwork( 3, 3 ), vtemp( 3 )

        IF( nstif .NE. 0 ) THEN
!..... 刚度阵坐标变换
          DO inode = 1, 2
            DO jnode = 1, 2
              iloca = ( inode - 1 ) * ndime
              jloca = ( jnode - 1 ) * ndime
              DO idime = 1, ndime
                DO jdime = 1, ndime
                  vwork( idime, jdime ) = 0.0D0
                  DO kdime = 1, ndime
                    idofe = iloca + idime
                    jdofe = jloca + kdime
                    vwork( idime, jdime ) = vwork( idime, jdime ) +          &
                    stife( idofe, jdofe ) * direc( jdime, kdime )
                  END DO
                END DO
              END DO
              DO idime = 1, ndime
                DO jdime = 1, ndime
                  idofe = iloca + idime
                  jdofe = jloca + jdime
                  stife( idofe, jdofe ) = 0.0D0
                  DO kdime = 1, ndime
                    stife( idofe, jdofe ) = stife( idofe, jdofe ) +          &
                    direc( idime, kdime ) * vwork( kdime, jdime )
                  END DO
                END DO
              END DO
            END DO
          END DO
        END IF

        IF( nforc .NE. 0 ) THEN
!..... 等效节点力坐标变换
          DO inode = 1, 2
            DO idime = 1, ndime
              vtemp( idime ) = 0.0D0
              DO jdime = 1, ndime
```

```
                idofe = ( inode - 1 ) * ndime + jdime
                vtemp( idime ) = vtemp( idime ) +                    &
                force( idofe ) * direc( idime, jdime )
              END DO
            END DO
            DO idime = 1, ndime
              idofe = ( inode - 1 ) * ndime + idime
              force( idofe ) = vtemp( idime )
            END DO
          END DO
        END IF
        RETURN
        END

        SUBROUTINE UpdateElast( elast, nlinc, iflag )
!.......
!       模块功能
!           确定 Kirchhoff 应力和 Green 应变间关系。
!.......
        USE CtrlData
        USE ElmtData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        IF( nlinc .EQ. 0 ) RETURN
        IF( iflag .EQ. 0 ) RETURN
!.....模量变换
        dprev = 0.0D0
        dcurr = 0.0D0
        DO idime = 1, ndime
          dtemp = coren( idime, 2 ) - coren( idime, 1 )
          dwork = dispe( ndofn + idime ) - dispe( idime ) + dtemp
          dprev = dprev + dtemp * dtemp
          dcurr = dcurr + dwork * dwork
        END DO

        ratio = DSQRT( dcurr / dprev )
!.....如果给出的是现实构形中模量或大变形小应变
        IF( iflag .EQ. 1 ) THEN
!.....割线模量
          elast = elast * 2.0D0 / ( ratio * ( ratio + 1.0D0 ) )
        ELSE
!.....切线模量
          elast = elast / ( ratio * ratio * ratio )
        END IF
        RETURN
```

END

例 8-1 假设杆发生弹性变形，且不考虑欧拉失稳，试计算图 8.2 所示结构顶铰竖向位移和荷载 P 的关系曲线。已知初始构形中杆件的长度为 L，面积为 A。

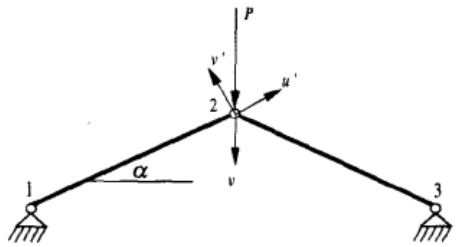


图8.2　二杆结构

记顶铰的竖向位移为 v，考虑杆件 1-2，端点 2 的轴向和法向位移分别记为 u' 和 v'，则

$$u' = -v\sin\alpha$$

$$v' = -v\cos\alpha$$

由于端点 1 的位移为 0，杆件 1-2 内的位移沿轴向线性分布，容易导出杆件内任意点的轴向 Green 应变为

$$E_{11} = \frac{\partial u'}{\partial x'} + \frac{1}{2}\left[(\frac{\partial u'}{\partial x'})^2 + (\frac{\partial v'}{\partial x'})^2\right] = -\frac{v}{L}\sin\alpha + \frac{1}{2}(\frac{v}{L})^2$$

轴向 Kirchhoff 应力为

$$S_{11} = E_S^0[-\frac{v}{L}\sin\alpha + \frac{1}{2}(\frac{v}{L})^2]$$

Kirchhoff 应力对应的伪轴向力为

$$N^0 = AS_{11} = E_S^0 A[-\frac{v}{L}\sin\alpha + \frac{1}{2}(\frac{v}{L})^2]$$

现时构形中的真实轴向力为

$$N = \frac{l}{L}N^0 = E_S^0 A\frac{l}{L}[-\frac{v}{L}\sin\alpha + \frac{1}{2}(\frac{v}{L})^2]$$

式中，l 为变形后杆件的长度，不难求得它与位移 v 间的关系式

$$l = L\sqrt{1 - 2\frac{v}{L}\sin\alpha + (\frac{v}{L})^2}$$

注意，现时构形中真实的轴向力是沿现时构形中杆件的轴线方向的。根据顶铰的平衡可得

$$2N\frac{L\sin\alpha - v}{l} + P = 0$$

将轴力 N 的表达式代入上式，得

$$\frac{v}{L}(\frac{v}{L} - \sin\alpha)(\frac{v}{L} - 2\sin\alpha) = \frac{P}{E_S^0 A} \tag{8.127}$$

将(8.125)式代入上式，得

$$2(\frac{v}{L} - \sin\alpha)(1 - \frac{1}{\sqrt{1 - 2\frac{v}{L}\sin\alpha + (\frac{v}{L})^2}}) = \frac{P}{EA} \tag{8.128}$$

定义无量纲位移，

$$\gamma = v/L\sin\alpha$$

则(8.127) 和(8.128)式可以改写为

$$\gamma(\gamma - 1)(\gamma - 2) = \frac{P}{E_S^0 A\sin^3\alpha} \tag{8.129}$$

$$\frac{2}{\sin^2\alpha}(\gamma-1)\left(1-\frac{1}{\sqrt{1-2\gamma\sin^2\alpha+\gamma^2\sin^2\alpha}}\right)=\frac{P}{EA\sin^3\alpha} \tag{8.130}$$

由上式可见，假定 E_S^0 和 E 为常数将得到不同的结果。为比较两者的区别，分别采用式(8.129)和(8.130)式分析 P 和 v 的关系，计算时假设(8.129)中的 E_S^0 和(8.130)式中的 E 为相同的常数，并定义无量纲荷载

$$\kappa=\frac{P}{EA\sin^3\alpha}=\frac{P}{E_S^0A\sin^3\alpha}$$

图 8.3 给出了 κ 随 γ 的变化曲线。由图可以看出，α 角不大或 $\gamma<2$ 时，假定 E_S^0 (为常数)与 E (为常数)相差不大。当位移相对较大时，两者间就有较大的区别，这一点应当引起重视。

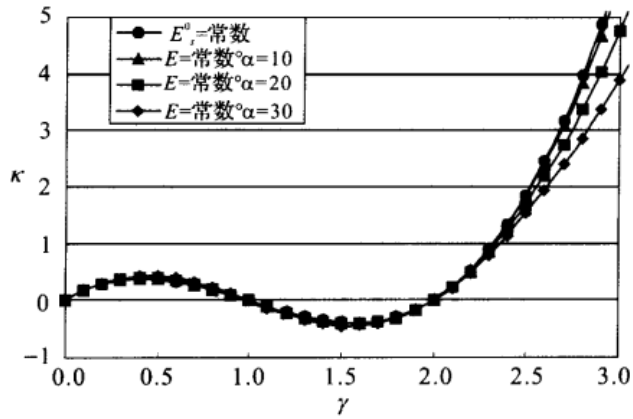


图 8.3 顶铰荷载位移曲线

例 8-2 为考察几何非线性和几何线性的区别，分析一长为 10 的平面杆件，一端简支，另一端自由。杆件截面面积为 0.0001，拉伸模量 E 为 2000000，质量密度为 100。初始时刻在自由端给定一初始速度(垂直于杆件的轴线方向)100。不考虑杆件的弯曲变形，整个杆件离散为一个杆单元。图 8.4 和 8.5 分别为几何线性和几何非线性条件下自由端的运行轨迹。由图可见，在

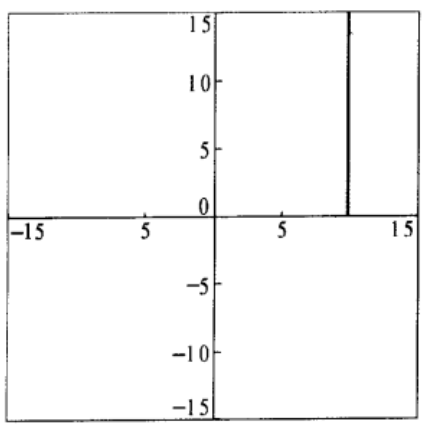


图 8.4 几何线性时自由端轨迹

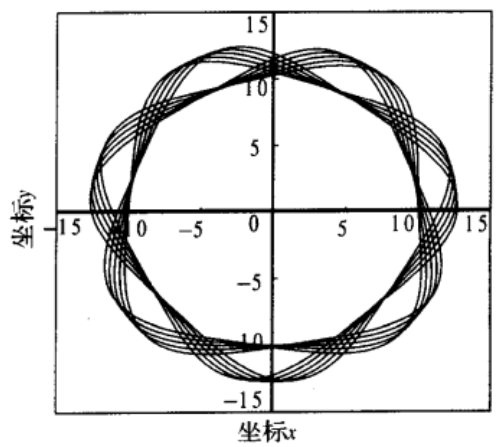


图 8.5 几何非线性时自由端轨迹

不考虑几何非线性的条件下，杆件将无限伸长，从而得到错误的结果。考虑几何非线性时，自由端的运动轨迹代表了两种运动规律：绕筒支端刚体转动和离心力引起的轴向振动。可见，不考虑几何非线性就不可能给出此类问题的正确描述。

8.5　2 节点大挠度平面梁单元

本节介绍大转动条件下 2 节点平面梁单元的切线刚度矩阵（Lagrange 法）。如图 8.6 示的 2 节点梁单元，以节点 1 为原点，单元轴向为 X_1' 轴建立直角坐标系 $OX_1'X_2'$。记沿 X_1' 和 X_2' 轴的位移为 u' 和 v'。用矢量形式表示为

$$\boldsymbol{u}' = \{u', v'\}^T \tag{8.131}$$

每个节点有 3 个独立的位移量：中性线上各点沿轴向位移 U'、挠度 V' 和转角 θ。单元节点位移矢量可表示为

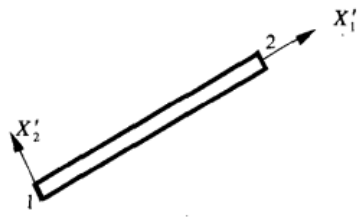


图 8.6　2 节点平面梁单元

$$\boldsymbol{d}'^{e} = \{U_1', V_1', \theta_1, U_2', V_2', \theta_2\}^T \tag{8.132}$$

单元内任意点的位移按下式插值得到

$$\boldsymbol{u}' = \boldsymbol{N}\boldsymbol{d}'^{e} \tag{8.133}$$

其中，

$$\boldsymbol{N} = \begin{bmatrix} 1-\xi & 6\xi(1-\xi)\dfrac{X_2'}{L} & -(1-\xi)(1-3\xi)X_2' & \xi & -6\xi(1-\xi)\dfrac{X_2'}{L} & \xi(2-3\xi)X_2' \\ 0 & 1-3\xi^2+2\xi^3 & L\xi(1-\xi)^2 & 0 & \xi^2(3-2\xi) & -L\xi^2(1-\xi) \end{bmatrix} \tag{8.134}$$

式中，L 为初始构形中单元的长度，$\xi = X_1'/L$。单元内的任意点的轴向 Green 应变为

$$E_{11} = \frac{\partial u_1'}{\partial X_1'} + \frac{1}{2}\left[\left(\frac{\partial u_1'}{\partial X_1'}\right)^2 + \left(\frac{\partial u_2'}{\partial X_1'}\right)^2\right] \tag{8.135}$$

大挠度大转角小应变条件下，忽略 $\left(\dfrac{\partial u_1'}{\partial X_1'}\right)^2$ 项，上式简化为

$$E_{11} = \frac{\partial u_1'}{\partial X_1'} + \frac{1}{2}\left(\frac{\partial u_2'}{\partial X_1'}\right)^2 \tag{8.136}$$

写成矩阵形式为

$$E_{11} = \boldsymbol{L}\ \boldsymbol{u}' + \frac{1}{2}\boldsymbol{A}\ \ \boldsymbol{H}\ \ \boldsymbol{u}' \tag{8.137}$$

其中，

$$\boldsymbol{L} = \left[\frac{\partial}{\partial X_1'},\ 0\right] \tag{8.138}$$

$$\boldsymbol{A} = \left[0,\ \frac{\partial v'}{\partial X_1'}\right] \tag{8.139}$$

$$\boldsymbol{H}=\begin{bmatrix}0 & 0\\ 0 & \dfrac{\partial}{\partial X_1'}\end{bmatrix} \tag{8.140}$$

将(8.134)式代入(8.137)式，可得

$$E_{11}=(\boldsymbol{B}_L+\frac{1}{2}\boldsymbol{A}\ \boldsymbol{G})\ \boldsymbol{d}'^{e}=\bar{\boldsymbol{B}}\boldsymbol{d}'^{e} \tag{8.141}$$

$$\bar{\boldsymbol{B}}=\boldsymbol{B}_L+\frac{1}{2}\boldsymbol{A}\ \boldsymbol{G} \tag{8.142}$$

其中，

$$\boldsymbol{B}_L=\frac{1}{L}[-1,\ 6(1-2\xi)\frac{X_2'}{L},\ 2(2-3\xi)X_2',\ 1,\ -6(1-2\xi)\frac{X_2'}{L},\ 2(1-3\xi)X_2'] \tag{8.143}$$

$$\boldsymbol{G}=\frac{1}{L}\begin{bmatrix}-1 & 6(1-2\xi)\dfrac{X_2'}{L} & 2(2-3\xi)X_2' & 1 & -6(1-2\xi)\dfrac{X_2'}{L} & 2(1-3\xi)X_2'\\ 0 & -6\xi(1-\xi) & L(1-\xi)(1-3\xi) & 0 & 6\xi(1-\xi) & L\xi(2-3\xi)\end{bmatrix} \tag{8.144}$$

将 Green 应变求变分，可得

$$\delta E_{11}=(\boldsymbol{B}_L+\boldsymbol{A}\boldsymbol{G})\,\delta\boldsymbol{d}'^{e}=\boldsymbol{B}\,\delta\boldsymbol{d}'^{e} \tag{8.145}$$

$$\boldsymbol{B}=\boldsymbol{B}_L+\boldsymbol{A}\ \boldsymbol{G} \tag{8.146}$$

设单元的轴向 Kirchhoff 应力 S_{11} 和 Green 应变间满足如下关系，

$$S_{11}=E\,E_{11} \tag{8.147}$$

将 (8.141)、(8.145)和(8.147)式代入虚功方程的左端可导得单元刚度矩阵，

$$\boldsymbol{K}'^{e}=\int_{V_0'}E\boldsymbol{B}^T\bar{\boldsymbol{B}}\mathrm{d}V \tag{8.148}$$

容易导出单元的切线刚度矩阵为

$$\boldsymbol{K}_T'^{e}=\boldsymbol{K}_L'+\boldsymbol{K}_S'+\boldsymbol{K}_N' \tag{8.149}$$

其中，

$$\boldsymbol{K}_L'=\begin{bmatrix}\dfrac{EA}{L} & 0 & 0 & \dfrac{EA}{L} & 0 & 0\\ & \dfrac{12EI}{L^3} & \dfrac{6EI}{L^2} & 0 & -\dfrac{12EI}{L^3} & \dfrac{6EI}{L^2}\\ & & \dfrac{4EI}{L} & 0 & -\dfrac{6EI}{L^2} & \dfrac{2EI}{L}\\ & & & \dfrac{EA}{L} & 0 & 0\\ & \text{对称} & & & \dfrac{12EI}{L^3} & -\dfrac{6EI}{L^2}\\ & & & & & \dfrac{4EI}{L}\end{bmatrix}$$

$$\boldsymbol{K}_s'=\frac{N}{L}\begin{bmatrix}1 & 0 & 0 & -1 & 0 & 0\\ & \frac{6}{5}+\frac{12I}{AL^2} & \frac{L}{10}+\frac{6I}{AL} & 0 & -(\frac{6}{5}+\frac{12I}{AL^2}) & \frac{L}{10}+\frac{6I}{AL}\\ & & \frac{2L^2}{15}+\frac{4I}{A} & 0 & -(\frac{L}{10}+\frac{6I}{AL}) & -\frac{L^2}{30}+\frac{2I}{A}\\ & & & 1 & 0 & 0\\ & \text{对称} & & & \frac{6}{5}+\frac{12I}{AL^2} & -(\frac{L}{10}+\frac{6I}{AL})\\ & & & & & \frac{2L^2}{15}+\frac{4I}{A}\end{bmatrix}$$

$$\boldsymbol{K}_N'=\frac{EA}{L^3}\begin{bmatrix}0 & \alpha_1 & \alpha_2 & 0 & -\alpha_1 & \alpha_3\\ & \beta_1 & \beta_2 & -\alpha_1 & -\beta_1 & \beta_3\\ & & \beta_4 & -\alpha_2 & -\beta_2 & \beta_6\\ & & & 0 & \alpha_1 & -\alpha_3\\ & \text{对称} & & & \beta_1 & -\beta_3\\ & & & & & \beta_7\end{bmatrix}$$

其中，

A — 初始构形中梁截面的面积；

I — 初始构形中梁截面对主轴的惯性矩。

$$\alpha_1=\frac{6L}{5}(v_2-v_1)-\frac{L^2}{10}(\theta_1+\theta_2)$$

$$\alpha_2=\frac{L^2}{10}(v_2-v_1)+\frac{L^3}{30}(\theta_2-4\theta_1)$$

$$\alpha_3=\frac{L^2}{10}(v_2-v_1)+\frac{L^3}{30}(\theta_1-4\theta_2)$$

$$\beta_1=\frac{72}{35}(v_2-v_1)^2+\frac{3L^2}{35}(\theta_1^2+\theta_2^2)-\frac{18L}{35}(v_2-v_1)(\theta_1+\theta_2)$$

$$\beta_2=\frac{9L}{35}(v_2-v_1)^2+\frac{L^3}{140}(\theta_2^2-\theta_1^2)-\frac{6L^2}{35}(v_2-v_1)\theta_1++\frac{L^3}{70}\theta_1\theta_2$$

$$\beta_3=\frac{9L}{35}(v_2-v_1)^2-\frac{L^3}{140}(\theta_2^2-\theta_1^2)-\frac{6L^2}{35}(v_2-v_1)\theta_2+\frac{L^3}{70}\theta_1\theta_2$$

$$\beta_4=\frac{3L^2}{35}(v_2-v_1)^2-\frac{L^3}{70}(v_2-v_1)(\theta_2-\theta_1)+\frac{L^4}{210}(12\theta_1^2+\theta_2^2-3\theta_1\theta_2)$$

$$\beta_5=-\frac{L^3}{70}(v_2-v_1)(\theta_1+\theta_2)-\frac{L^4}{420}(3\theta_1^2+3\theta_2^2-4\theta_1\theta_2)$$

$$\beta_6=\frac{3L^2}{35}(v_2-v_1)^2+\frac{L^3}{70}(v_2-v_1)(\theta_2-\theta_1)+\frac{L^4}{210}(\theta_1^2+12\theta_2^2-3\theta_1\theta_2)$$

(8.149)式经过坐标变换就可得到整体坐标系下的单元切线刚度矩阵。

例8-3 随动轴向载荷作用下柱体受初始横向挠动的强迫振动。

如图8.7所示，一端固定的柱体受轴向压力 P 作用，假定压力 P 始终作用在柱体的轴线方向，是随动载荷。P_1 为水平方向的初始挠动，它使柱体产生横向振动。柱体振动的频率和幅值的大小将随轴向力 P 的改变而改变。Bernuli梁的动屈曲临界载荷的解析解为

$$P_{cr}=\frac{2.008\pi EI}{L^2}$$

式中，EI 为梁的弯曲刚度，L 为梁的长度。

取柱体的长度 $L=1000\text{mm}$，宽度 b 为单位长度。梁高 $h=2.289\text{mm}$，材料的杨氏模量为 $E=1\times10^5\text{MPa}$，泊松比 $\nu=0.25$，单位长度的质量密度为 $2\times10^{-7}\text{kg/mm}^3$。用10个梁单元来模拟柱体的振动。

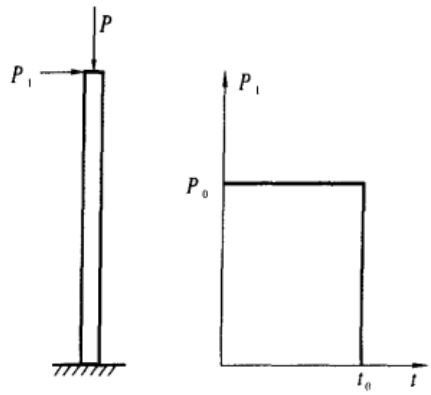


图8.7 轴压作用下的柱体

图8.8~8.11为不同横向挠动作用下自由端挠度的时间历程图，图8.12为不同横向挠动作用下8秒内自由端最大挠度与轴向力 P 的关系曲线。

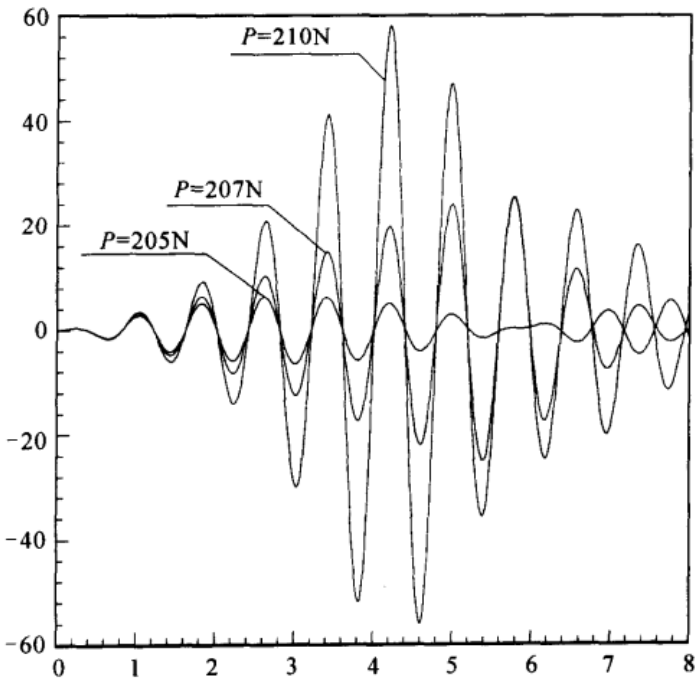


图8.8 $P_1=0.1\text{N}$ 临界载荷附近自由端挠度(mm)-时间(s)曲线

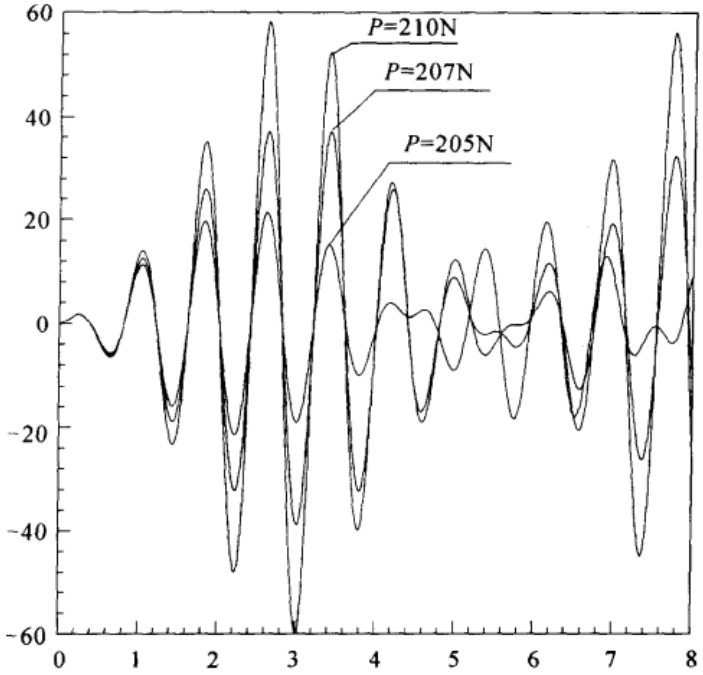


图8.9　$P_1 = 0.4$N 临界载荷附近自由端挠度(mm)-时间(s)曲线

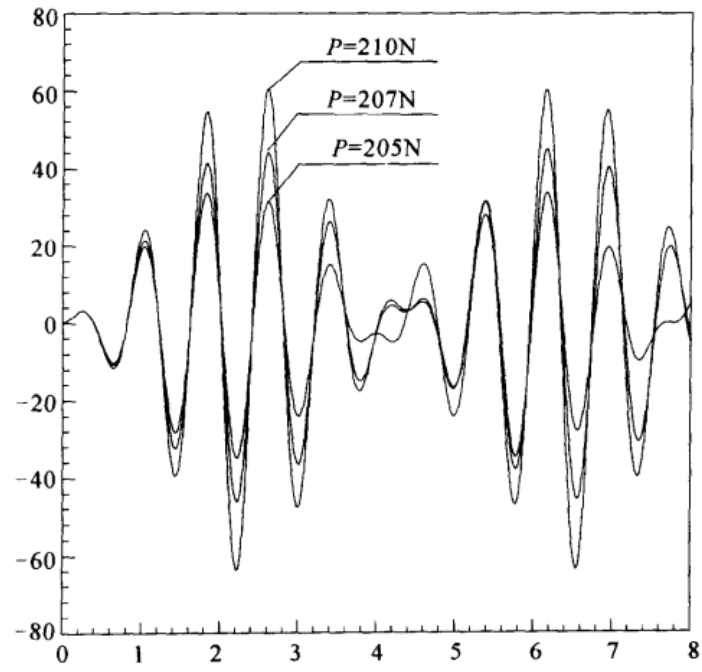


图8.10　$P_1 = 0.7$N 临界载荷附近自由端挠度(mm)-时间(s)曲线

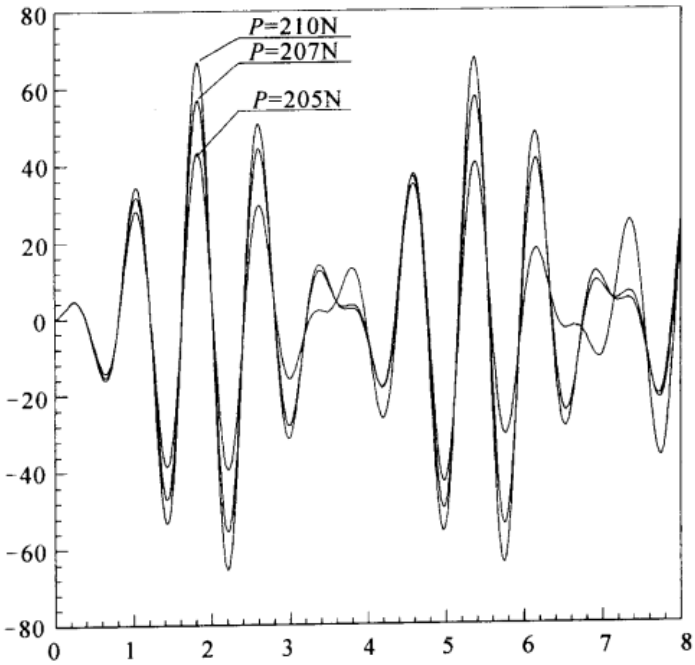


图8.11 $P_1=1.0$N 临界载荷附近自由梁端挠度(mm)-时间(s)曲线

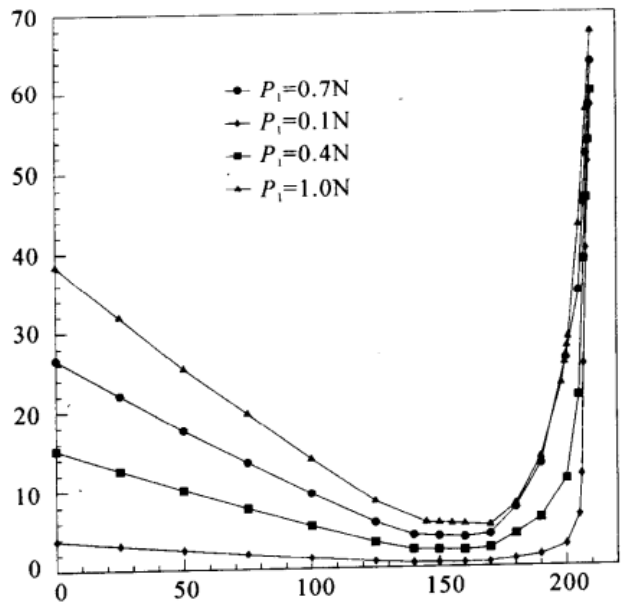


图8.12 8秒内自由端最大挠度(mm)-P(N)曲线

由图8.12可以看出，随轴向载荷的增加，端点横向自由振动的最大幅值比较缓慢地减少，但是当载荷接近临界值时，最大幅值将随轴向载荷的增加而迅速增加。

8.6　非线性自由振动的迭代响应法

与线性系统的自由振动相比，非线性自由振动具有本质的区别，后者更为复杂，在工程实践中更常见。因此，非线性系统的自由振动问题一直受到人们的关注。

一、平均迭代法和平均刚度法

国内外众多学者在这方面做了不少的研究工作，提出了不少分析方法。目前比较常用的分析方法有两种。

第一种方法是假定模态法，即先假定自由振动的空间分布模式(通常为一阶线性模态或若干低阶线性模态的迭加)，并在一定振幅下计算刚度矩阵，进而计算自由振动的频率，或者采用Galerkin法求解非线性微分方程，通常这类微分方程都是Duffing型的。我们称这种方法为平均迭代法。采用假定模态法，C.L.Lou等[1]和W.Y.Tseng等[2]分析了简支梁非线性自由振动问题。J.A.Bennett和J.G..Eisley[3]采用三阶线性模态和Galerkin法分析了夹支梁自由振动问题；H.R.Busby和V.I.Weingarten[4]采用有限单元法综合两阶模态分析了一端固支一端简支梁的自由振动；C. Mei[5]采用有限元法分析了各种边界条件下梁的自由振动和简谐激励下的响应；C.L.D. Huang和S.T. Huang[6]采用有限单元法分析了中心带有刚性质量块圆板的非线性轴对称自由振动；江晓禹等[7]分析了圆柱壳的非线性自由振动。

另一种方法则是事先假定自由振动的某种谐振方式（但不一定是简谐振动形式），然后采用某种方法(如Kantornovich时间平均法)得到一个周期内的平均刚度，从而确定自由振动频率和振幅之间的关系。我们称这种方法为平均刚度法。C.L. Huang和H.S. Walker[8]采用这一方法研究了中心带有刚性质量块圆板的非线性轴对称自由振动；任宝生和陈殿云等[9]采用有限元法分析了同一问题。

我们知道，线性系统自由振动的任何时刻各位移分量的相对值保持不变，即空间分布模式不变。在整个自由振动过程中，任意时刻的瞬时角频率保持不变。而非线性系统则不具备这两个性质，刚阵 $\boldsymbol{K}$ 中的非线性部分间接或直接地与时间有关，因此自由振动过程中位移的空间分布模式与过程有关。从而导致了自由振动的频率与振幅有关，同时，线性系统的简谐振动假定也不再成立。由此可见，假定模态和假定谐振方式都有一定的局限性，没能很好地反映非线性自由振动的本质。

为说明问题，我们考察如下具有对称恢复力的单自由度系统，

$$\ddot{x} + \omega_0^2(x + \varepsilon x^3) = 0$$

运用Lindstedt-Poincare方法、平均刚度法和平均迭代法求上述问题，分别得到非线性振动的角频率 ω 和幅值 A 的关系的二次近似解，

$$\frac{\omega^*}{\omega_0} = 1 + \frac{3}{8}(\varepsilon A^2) - \frac{21}{256}(\varepsilon A^2)^2$$

$$\frac{\omega}{\omega_0} = \sqrt{1 + \frac{3}{4}\varepsilon A^2} \approx 1 + \frac{3}{8}(\varepsilon A^2) - \frac{18}{256}(\varepsilon A^2)^2$$

$$\frac{\omega}{\omega_0} = \sqrt{1 + \varepsilon A^2} \approx 1 + \frac{1}{2}(\varepsilon A^2) - \frac{1}{8}(\varepsilon A^2)^2$$

比较上述三个解不难看出，理论解比平均刚度法和平均迭代法的解都小。相对而言，平均刚度法比平均迭代法具有更高阶的精度。随振幅A的增加，平均迭代法和平均刚度法导致的误差越来越大。

由于与过程相关，非线性结构自由振动问题只能通过对自由振动过程的模拟来研究。本节介绍的迭代响应方法就是，通过对恰当初值问题的响应分析来研究非线性结构自由振动频率和振幅间的关系的。

二、初值问题及迭代方法

不考虑阻尼的作用，即假设$\boldsymbol{C}=\boldsymbol{0}$，此时非线性有限元离散系统自由振动的动力学平衡方程可表示为

$$\boldsymbol{M}\ddot{\boldsymbol{d}}+\boldsymbol{K}\boldsymbol{d}=\boldsymbol{0} \tag{8.150}$$

对上式关于时间t求全微分，得

$$\boldsymbol{M}\dot{\boldsymbol{v}}+\boldsymbol{K}_T\boldsymbol{v}=\boldsymbol{0} \tag{8.151}$$

式中，$\boldsymbol{v}=\dot{\boldsymbol{d}}$为速度矢量，$\boldsymbol{K}_T$为系统的切向刚度矩阵。假定系统自由振动时所有自由度同时通过系统平衡位置$\boldsymbol{d}=\boldsymbol{0}$，而且在系统平衡位置附近，$\boldsymbol{K}_T$变化很小。由上式可以求得速度的归一化模态$\boldsymbol{\varphi}$。注意，这里的归一化指的是令$\boldsymbol{\varphi}$中的最大分量为1。将$\boldsymbol{\varphi}$看作为非线性系统自由振动经过平衡位置时速度矢量的近似值，构造以下初值问题，

$$\begin{cases}\boldsymbol{M}\ddot{\boldsymbol{d}}+\boldsymbol{K}\boldsymbol{d}=\boldsymbol{0}\\ \boldsymbol{d}\,|_{t=0}=\boldsymbol{0}\\ \dot{\boldsymbol{d}}\,|_{t=0}=v_{\max}\boldsymbol{\varphi}\end{cases} \tag{8.152}$$

式中，$v_{\max}$为速度因子，不同$v_{\max}$将对应不同的振动幅值。取$\boldsymbol{\varphi}$中具有最大分量的自由度作为参考自由度，记参考自由度振动一个周期回到零点的时间为T，则$1/T$可作为非线性系统自由振动频率的近似值。事实上，上述初值问题并不能保证所有的自由度历经一个周期后同时回到平衡位置，即T和非线性周期之间还是有一定的误差。为提高计算精度，构造响应计算的迭代格式，我们称之为迭代响应分析法。迭代响应分析法的具体步骤如下。

(1) 求解对应线性系统的特征值问题：

$$(\boldsymbol{K}\,|_{d=0}-\omega^2\boldsymbol{M})\boldsymbol{\varphi}=\boldsymbol{0}$$

得某阶频率ω及其对应的归一化模态$\boldsymbol{\varphi}$。

(2) 选择参考自由度和速度因子$v_{\max}$。

(3) 计算初值问题的响应。

(4) 按参考自由度计算非线性频率的近似值。

(5) 判断迭代频率是否收敛，如果收敛，输出结果，停机。否则进行下一步。

(6) 计算

$$\boldsymbol{\varphi}=\frac{v_{\max}\boldsymbol{\varphi}+\bar{\boldsymbol{v}}}{\max(v_{\max}\boldsymbol{\varphi}+\bar{\boldsymbol{v}})}$$

式中，$\bar{\boldsymbol{v}}$为参考自由度回到平衡位置时的速度分布，转(3)。

数值计算表明，上述平均迭代法具有很好的收敛性。迭代次数与非线性强弱有关。对几何非线性系统，振幅越大需要的迭代步数越多。一般迭代两次即可得到较好的结果，迭代三到五

次即收敛。表8.1给出了一固支梁(长度为100mm，厚度为1mm，考虑几何非线性)自由振动迭代分析时的迭代过程。

表8.1　固支梁自由振动迭代分析

迭代次数	0	1	2	3	4
频率(Hz)	16.950	26.066	25.922	25.928	25.928
幅值(mm)	线　性	6.467	6.405	6.407	6.407

例8-4 中心附有刚体质量的均匀圆板。

为验证方法的适用性，考虑如图8.13所示等厚度圆板，板的外半径为 a，内半径为 b，板的厚度为 h，圆板中心附有刚体质量 M_c，刚体质量块的半径为圆板的内半径。文献[6][8][9][10]都曾对这一问题进行了探讨。

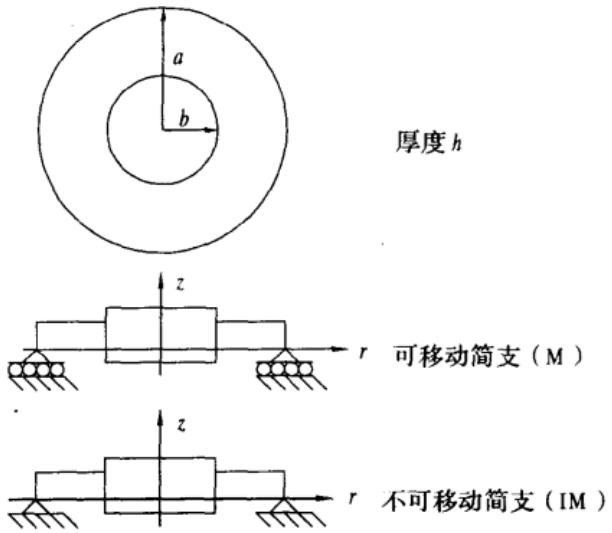


图8.13　中心附有刚体质量的均匀圆板

设圆板的质量密度为 ρ，杨氏模量为 E 和泊松比为 ν。引入无量纲量

$$R=\frac{b}{a}\qquad \gamma=\frac{M_c}{\pi b^2\rho h}$$

轴对称环板单元考虑了中等转动引入的几何非线性，按本节提供的方法计算得到的数值结果与文献[8][10]的结果比较如图8.14~8.16所示。图8.14、8.15和8.16分别为 $\gamma=1.5$、$\gamma=2.0$ 和 $\gamma=3.0$ 时非线性自由振动频率的计算结果。图中纵轴为板内缘振幅 w_R 和板厚度 h 的比值，横轴为非线性自由振动的基频，M表示可移动简支，IM表示不可移动简支。为图形清晰起见，图中没有列出文献[6]和文献[9]的结果。

由图可见，无论可移动简支还是不可移动简支，频率都随振幅的增大而增高，圆板-刚性质量块系统表现出类似duffing方程的硬弹簧特性。刚性质量的大小对板自由振动的频率有很大的影响，较大的 γ 对应较低的基频；同时，刚性质量的大小对非线性性质也有很大的影响，质量块越小，随振幅增加频率增加越快，因而具有较强的非线性。影响系统非线性性质的另外一

个重要因素是边界条件，由图可见，不可移动简支边界导致较强的非线性，而可移动简支边界对应弱非线性。

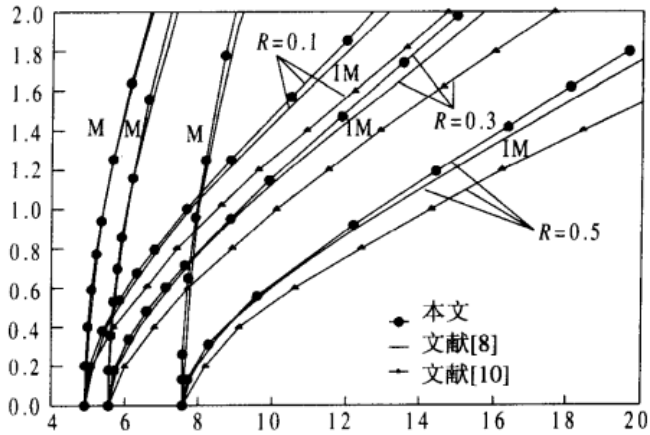


图8.14 非线性自由振动频率（$\gamma=1.5$）

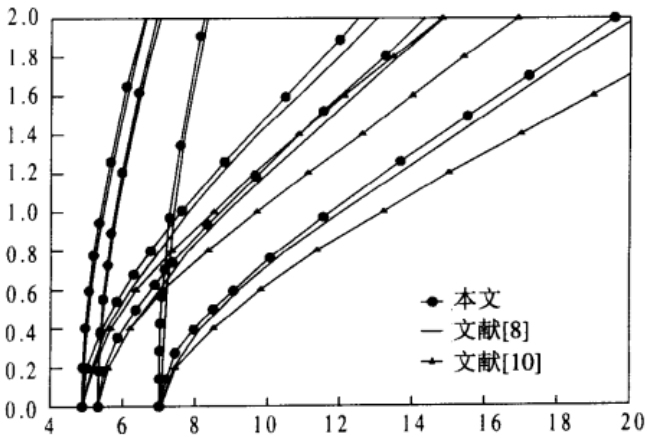


图8.15 非线性自由振动频率（$\gamma=2.0$）

（注：图中标注参见图8.14）

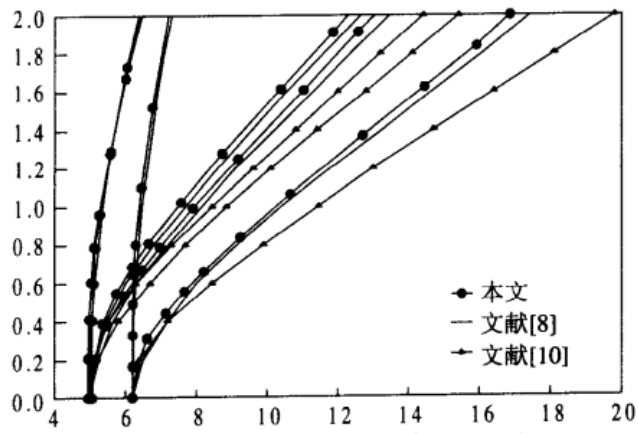


图8.16 非线性自由振动频率（$\gamma=3.0$）

（注：图中标注参见图8.14）

由图还可以看到，三种方法的数值结果具有相同的趋势，当振幅很小时，三种方法的结果吻合得很好，随振幅的增加，三种结果的差别逐渐增加，特别是文献[10]的数值结果。与文献[8]比较，文献[10]的频率较高，本文计算得到的基频略低于文献[8]，三种结果的相互关系和单自由度分析的结果完全相同。

参考文献

[1]C.L.Lou and D.L.Sikarskie. Non-linear vibrations of beam using a form-function approximation. *Journal of Applied Mechanics*, 1975, 42

[2]W.Y.Tseng and J. Dugundji. Non-linear vibrations of a beam under harmonic excitation. *Journal of Applied Mechanics*, 1971, 38

[3]J.A. Bennett and J.G. Eisley. A multiple degree-of-freedom approach to non-linear beam vibrations. *American Institute of Aeronautics and Astronautics Journal*, 1970, 8

[4]H.R.Busby and V.I.Weingarten. Non-linear response of a beam to periodic loading. *International Journal of Non-linear Mechanics*, 1972, 7

[5]C.Mei and K.Decha-Umphai. A finite element method for non-linear forced vibrations of beams. *Journal of Sound and Vibration*, 1985, 102(3)

[6]C.L.D.Huang and S.T.Huang. Finite Element Analysis of Non-Linear Vibration of a Circular Plate with a Concentric Rigid Mass. *J. of Sound and Vibration*, 1989, 131(2)

[7]江晓禹，张相周.圆柱壳的非线性自由振动分析.应用力学学报，1993, 10(3)

[8]C.L.D.Huang and H.S.Walker. Non-linear Vibration of a Hinged Circular Plate with a Concentric Rigid Mass. *Journal of Sound and Vibration*，1988, 126(1)

[9]任宝生，陈殿云，张以忠.计算圆板大幅度非线性振动频率的平均刚度法.计算结构力学及其应用，1995, 12(2)

[10]陈殿云，任宝生.圆板非线性有限元分析的一种迭代方法.应用力学学报，1993, 10(3)

第9章

材料非线性问题的有限单元法

本章介绍小变形材料非线性问题的有限单元法。从相对简单的非线性弹性问题开始，介绍常用的直接迭代法、切线刚度法和初应力法等在材料非线性有限元分析中的应用。重点介绍了弹塑性问题有限单元法求解过程中的一些技术细节，并给出了小变形平面和空间节理单元的有限元列式，最后给出了部分弹塑性有限元分析的共用程序模块。

9.1 非线性弹性问题的有限单元法

小变形非线性弹性问题是所有材料非线性问题中最简单的一类问题，与线弹性问题的有限单元法相比，两者的不同只表现在应力应变关系上，许多线弹性问题的有限元列式这里仍然适用。

对于非线性弹性问题，应力和应变间仍然存在一一对应关系，但不再保持线性。工程上常用的非线性弹性本构关系常常是根据材料力学试验拟合得到的，材料常数即为各个拟合参数。不同的材料可以采用不同的拟合模型和相应的拟合参数。例如，金属材料单向拉伸试验得到的 Romberg-Osgood 模型为

$$\varepsilon = \frac{\sigma}{E} + k(\frac{\sigma}{E})^n \tag{9.1}$$

式中，E 为初始模量，k, n 为两个拟合试验参数。又如在岩土工程中常用的 Duncan-Zhang 模型就是用双曲线拟合三轴压缩试验时的应力应变曲线得到的。除曲线拟合方法外，通过插值或其他数学方法也可以得到材料的非线性应力应变关系。

由于是小变形问题，线性的几何关系依然成立，

$$\boldsymbol{\varepsilon} = \boldsymbol{Bd} \tag{9.2}$$

为方便起见，上式中我们省略了代表单元的上标 e。由虚功方程不难得到如下形式的平衡方程，

$$\int_V \boldsymbol{B}^T \boldsymbol{\sigma} \mathrm{d}V = \boldsymbol{F} \tag{9.3}$$

必须注意，由于小变形的关系，应力形式的平衡方程(9.3)仍然是线性的，但是以节点位移表示的平衡方程不再是线性的了，因为应力应变关系是非线性的，从而应力和位移之间也是非线性的。将应力应变关系代入(9.3)式可得到位移表示的平衡方程，

$$\boldsymbol{K}(\boldsymbol{d})\boldsymbol{d} = \boldsymbol{F} \tag{9.4}$$

求解以上非线性方程组的方法很多，在第 2 章中也有介绍。以下叙述其中几种常用的方法在非线性弹性问题中的应用，部分方法可以直接推广到其他材料非线性问题的求解。

9.1.1 直接迭代法

直接迭代法也叫割线刚度法。如果材料的应力应变关系能够表示成，

$$\boldsymbol{\sigma} = \boldsymbol{D}_S(\boldsymbol{\varepsilon})\boldsymbol{\varepsilon} \tag{9.5}$$

将(9.2)式代入上式，

$$\boldsymbol{\sigma} = \boldsymbol{D}_S(\boldsymbol{\varepsilon})\boldsymbol{B}\boldsymbol{d}$$

再将上式代入(9.3)式，得

$$\int_V \boldsymbol{B}^T\boldsymbol{\sigma}\mathrm{d}V = \int_V \boldsymbol{B}^T\boldsymbol{D}_S(\boldsymbol{\varepsilon})\boldsymbol{B}\mathrm{d}V\boldsymbol{d} = \boldsymbol{K}_S(\boldsymbol{d})\boldsymbol{d} = \boldsymbol{F} \tag{9.6}$$

式中，

$$\boldsymbol{K}_S = \int_V \boldsymbol{B}^T\boldsymbol{D}_S(\boldsymbol{\varepsilon})\boldsymbol{B}\mathrm{d}V$$

$\boldsymbol{K}_S$ 为割线刚度矩阵，它在形式上和线性有限单元法的刚度矩阵没有什么区别。利用(9.6)式可构造出如下直接迭代格式，

$$\boldsymbol{K}(\boldsymbol{d}^{(k)})\boldsymbol{d}^{(k+1)} = \boldsymbol{F} \tag{9.7}$$

具体迭代步骤如下，

(1) 取迭代初值 $\boldsymbol{d}^{(0)} = \boldsymbol{0}$，$k = 0$；

(2) 计算当前迭代步的割线刚度矩阵 $\boldsymbol{K}^{(k)} = \boldsymbol{K}(\boldsymbol{d}^{(k)})$；

(3) 计算位移 $\boldsymbol{d}^{(k+1)} = (\boldsymbol{K}^{(k)})^{-1}\boldsymbol{F}$；

(4) 判断是否收敛，如果收敛，停止迭代；否则，令 $k = k+1$，转步骤(2)进行下一步迭代计算。

采用直接迭代法求解方程(9.6)式时，要注意两个事实：一是直接迭代法的每次迭代都要重新计算刚度阵，是一种变刚度法，计算量较大；二是直接迭代法收敛速度慢，需要的迭代次数多，且不一定能保证收敛。因此，在分析具体问题时需要根据问题的特点设定一个最大的迭代步数。迭代收敛的判断标准可采用第 2 章建议的方法。

迭代过程中单元应力的变化如图 9.1 所示。

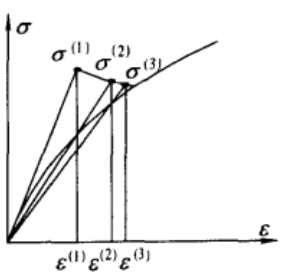


图 9.1 直接迭代法单元应力

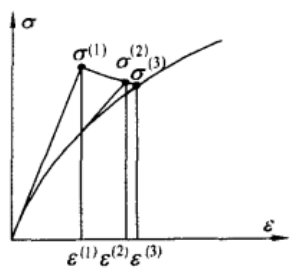


图 9.2 切线刚度法单元应力

9.1.2 切线刚度法

如果材料的应力应变关系能够采用增量形式表示成，

$$\mathrm{d}\boldsymbol{\sigma} = \boldsymbol{D}_T(\boldsymbol{\varepsilon})\mathrm{d}\boldsymbol{\varepsilon} \tag{9.8}$$

式中，$\boldsymbol{D}_T$ 是切线弹性矩阵。根据 Newton-Raphson 法的思想，并注意到 $\mathrm{d}\boldsymbol{\varepsilon} = \boldsymbol{B}\mathrm{d}\boldsymbol{d}$，(9.4)式可按如下格式迭代计算，

$$\boldsymbol{K}_T^{(k)}\Delta\boldsymbol{d}^{(k)} = \boldsymbol{F} - \int_V \boldsymbol{B}^T\boldsymbol{\sigma}^{(k)}\mathrm{d}V = \boldsymbol{R}^{(k)} \tag{9.9}$$

$$\boldsymbol{K}_T = \frac{\partial}{\partial \boldsymbol{d}}\left(\int_V \boldsymbol{B}^T\boldsymbol{\sigma}\mathrm{d}V - \boldsymbol{F}\right) = \int_V \boldsymbol{B}^T\boldsymbol{D}_T\boldsymbol{B}\mathrm{d}V \tag{9.10}$$

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} + \Delta\boldsymbol{d}^{(k)} \tag{9.11}$$

$$\boldsymbol{d}^{(0)} = \boldsymbol{0} \tag{9.12}$$

式中，$\boldsymbol{K}_T$ 为切线刚度矩阵，$\boldsymbol{R}^{(k)}$ 为第 k 次迭代得到的应力场对应的失衡力。具体迭代步骤如下，

(1) 取迭代初值 $\boldsymbol{d}^{(0)} = \boldsymbol{0}$，$k = 0$；

(2) 计算当前迭代步的切线刚度矩阵 $\boldsymbol{K}_T^{(k)} = \boldsymbol{K}_T(\boldsymbol{d}^{(k)})$；

(3) 计算残差 $\boldsymbol{R}^{(k)}$；

(4) 计算位移增量和位移 $\Delta\boldsymbol{d}^{(k)} = (\boldsymbol{K}_T^{(k)})^{-1}\boldsymbol{R}^{(k)}$，$\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} + \Delta\boldsymbol{d}^{(k)}$；

(5) 判断是否收敛，如果收敛，停止迭代；否则，令 $k = k+1$，转步骤(2)进行下一步迭代计算。

切线刚度法也是一种变刚度法，迭代过程中单元应力的变化如图 9.2 所示。

9.1.3 应力转移法和初应力法

与直接迭代法相比，切线刚度法一般具有较快的收敛速度，但是，它的每次迭代仍然需要重新计算总体切线刚度阵，并求解一个线性方程组。如果将(9.9)式中的 $\boldsymbol{K}_T^{(k)}$ 一律改成 $\boldsymbol{K}_T^{(0)}$，则迭代公式变为

$$\boldsymbol{K}_T^{(0)}\Delta\boldsymbol{d}^{(k)} = \boldsymbol{F} - \int_V \boldsymbol{B}^T\boldsymbol{\sigma}^{(k)}\mathrm{d}V = \boldsymbol{R}^{(k)} \tag{9.13}$$

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} + \Delta\boldsymbol{d}^{(k)} \tag{9.14}$$

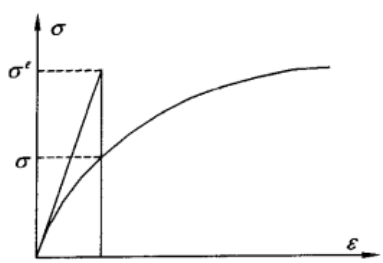


图 9.3 “初应力”含义

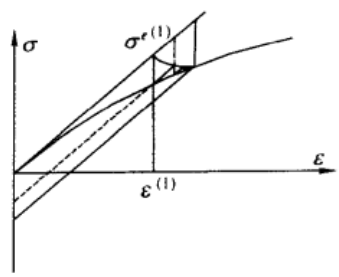


图 9.4 初应力迭代法单元应力

由于 $\boldsymbol{R}^{(k)}$ 是第 k 次迭代得到的应力场对应的失衡力，按(9.13)和(9.14)式迭代一次，相当于将失衡力按弹性方式重新分配到各个节点。这样做势必夸大了部分单元的刚度，使它们承受了过大的应力，也可能低估了部分单元的刚度，从而没有充分发挥它们承受荷载的能力。这种按线性弹性计算的应力与材料实际所承受的应力之差的等效节点力就是这一步计算对应的

失衡力，将在下一步迭代中得到重新分配。因此，Zienkiewicz 最初提出这种方法时，称之为应力转移法。

将(9.14)式代入(9.13)式，得到如下形式的迭代公式，

$$\boldsymbol{K}_T^{(0)}\boldsymbol{d}^{(k+1)} = \boldsymbol{F} - \int_V \boldsymbol{B}^T(\boldsymbol{\sigma}^{(k)} - \boldsymbol{\sigma}^{e(k)})\mathrm{d}V \tag{9.15}$$

$$\boldsymbol{\sigma}^{e(k)} = \boldsymbol{D}_T^0\boldsymbol{\varepsilon}^{(k)} \tag{9.16}$$

上式推导时利用了如下等式，

$$\boldsymbol{K}_T^{(0)}\boldsymbol{d}^{(k)} = \int_V \boldsymbol{B}^T\boldsymbol{D}_T^0\boldsymbol{B}\mathrm{d}V\boldsymbol{d}^{(k)} = \int_V \boldsymbol{B}^T\boldsymbol{D}_T^0\boldsymbol{\varepsilon}^{(k)}\mathrm{d}V = \int_V \boldsymbol{B}^T\boldsymbol{\sigma}^{e(k)}\mathrm{d}V \tag{9.17}$$

注意，$\boldsymbol{\sigma}^{e(k)}$ 是按线弹性本构关系计算出的应力矢量，简称弹性应力，它不是真实存在的。设迭代过程收敛时，$\boldsymbol{\sigma}^{(k)}$ 和 $\boldsymbol{\sigma}^{e(k)}$ 分别收敛于 $\boldsymbol{\sigma}$ 和 $\boldsymbol{\sigma}^e$。此时，$\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \boldsymbol{\sigma}^e$ 并不是零矢量，我们可以把它看成是等价线性结构具有的“初应力”，其含义如图 9.3 所示。迭代过程(9.17)实际上相当于在寻找这个“初应力”矢量，所以这种迭代方法称为初应力法。当然，这个“初应力”也是一个虚拟的初应力，它并不真实存在。由于迭代过程中刚度矩阵保持不变，因此这种方法也称为等刚度法。初应力法迭代过程中单元应力的变化如图 9.4 所示。

9.1.4 初应变法

相应地，失衡力也可以用初应变表示。将应变用应力表示，物理方程可以写成

$$\boldsymbol{\varepsilon} = \boldsymbol{C}(\boldsymbol{\sigma})\boldsymbol{\sigma} \tag{9.18}$$

类比初应力法，设想存在一个虚拟的初应变 $\hat{\boldsymbol{\varepsilon}}$，使得应力应变可以用如下线弹性物理方程来表示，

$$\boldsymbol{\sigma} = \boldsymbol{D}_T^{(0)}(\boldsymbol{\varepsilon} - \hat{\boldsymbol{\varepsilon}}) \tag{9.19}$$

联立(9.18)和(9.19)式有

$$\hat{\boldsymbol{\varepsilon}} = [\boldsymbol{C}(\boldsymbol{\sigma}) - (\boldsymbol{D}_T^{(0)})^{-1}]\boldsymbol{\sigma} \tag{9.20}$$

如果引入假想的弹性应变

$$\boldsymbol{\varepsilon}^e = (\boldsymbol{D}_T^{(0)})^{-1}\boldsymbol{\sigma} \tag{9.21}$$

则有

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^e \tag{9.22}$$

如图 9.5 所示。

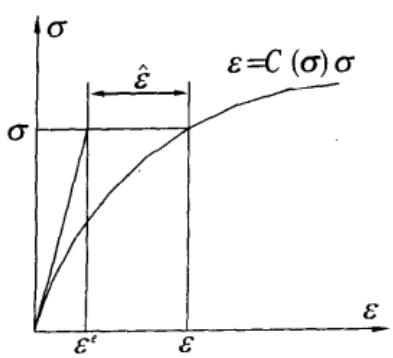


图 9.5 “初应变”含义

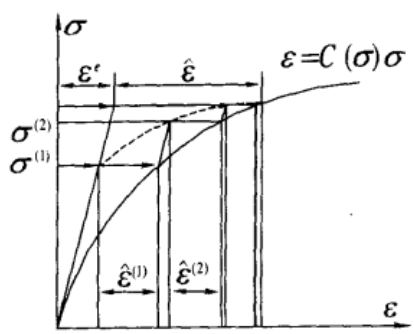


图 9.6 初应变迭代法单元应力

把(9.19)式代入(9.3)式，有

$$\boldsymbol{K}_T^{(0)}\boldsymbol{d}=\boldsymbol{F}+\int_V \boldsymbol{B}^T\boldsymbol{D}_T^{(0)}\hat{\boldsymbol{\varepsilon}}\mathrm{d}V \tag{9.23}$$

可以把上式写成如下迭代格式，

$$\boldsymbol{K}_T^{(0)}\boldsymbol{d}^{(k+1)}=\boldsymbol{F}+\boldsymbol{F}^{(k)} \tag{9.24}$$

$$\boldsymbol{F}^{(k)}=\int_V \boldsymbol{B}^T\boldsymbol{D}_T^{(0)}\hat{\boldsymbol{\varepsilon}}^{(k)}\mathrm{d}V \tag{9.25}$$

式中，称 $\boldsymbol{F}^{(k)}$ 为矫正荷载，它近似地等于抵消初应变所需要的荷载。整个迭代过程就是一个调整“初应变”的过程，应此称之为初应变法。整个迭代过程中单元内应力的变化如图 9.6 示。

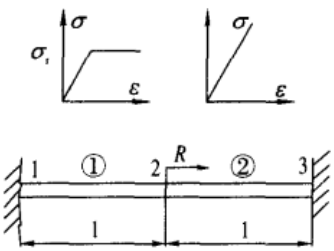


图9.7 简单杆件

例 9-1 采用直接迭代法、切线刚度法和初应力法计算如图 9.7 所示杆件。已知杆长 $2l$，截面面积为 A。杆左半部分材料满足理想弹塑性应力应变关系，屈服应力为 σ_s，弹性模量为 E。杆右半部分满足线弹性应力应变关系，弹性模量也为 E。中点作用的荷载为 $R=3A\sigma_s$。

将杆系统离散为两个杆单元，如图 9.7 所示。节点位移分别记为 u_1、u_2 和 u_3。有限元系统的节点位移矢量为 $\boldsymbol{d}=\{u_1,\ u_2,\ u_3\}^T$。根据位移约束条件有 $u_1=u_3=0$。单元①和②的应力和应变分别记为 σ_1，ε_1，σ_2，ε_2。则

$$\varepsilon_1=-\varepsilon_2=\frac{u_2}{l}$$

一、直接迭代法

系统的整体割线刚度矩阵为

$$\boldsymbol{K}=\frac{EA}{l}\begin{bmatrix}\zeta & -\zeta & \\ -\zeta & 1+\zeta & -1\\ & -1 & 1\end{bmatrix}\qquad \zeta=\frac{E_s}{E}$$

式中，E_s 为单元①的割线拉伸模量，是应变的函数。引入约束条件后，有限元离散系统的平衡方程为

$$(1+\zeta)\frac{EA}{l}u_2=3A\sigma_s$$

上式可进一步简化为

$$\bar{u}_2=\frac{3}{(1+\zeta)}$$

其中，

$$\bar{u}_2=\frac{E}{\sigma_s l}u_2$$

迭代公式为

$$\bar{u}_2^{(k+1)}=\frac{3}{(1+\zeta^{(k)})}\qquad \bar{u}_2^{(0)}=0$$

迭代计算结果列于表 9.1 中，迭代过程中应力的变化如图 9.8 所示。表中变量为

$$\bar{\sigma}_1=\frac{\sigma_1}{\sigma_s}\qquad\qquad \bar{\sigma}_2=\frac{\sigma_2}{\sigma_s}$$

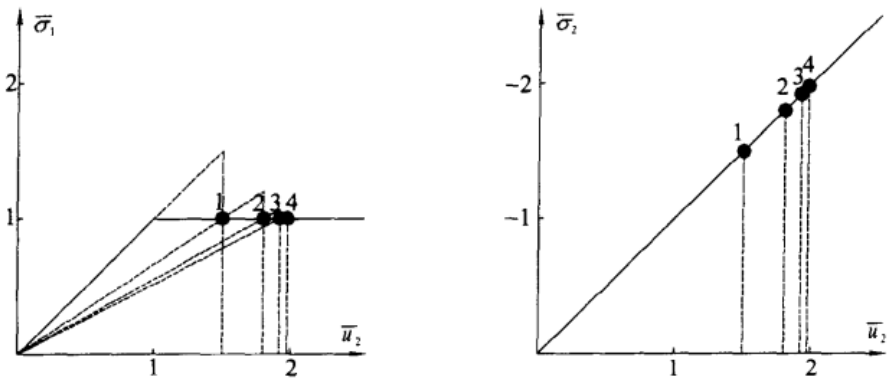


图 9.8 直接迭代法单元应力变化

表 9.1 直接迭代法迭代过程

k	0	1	2	3	4	5
$\bar{u}_2^{(k+1)}$	0	3/2	9/5	27/14	81/41	243/122
ζ	1	2/3	5/9	14/27	41/81	122/243
$\bar{\sigma}_1$	0	1	1	1	1	1
$\bar{\sigma}_2$	0	-3/2	−9/5	−27/14	−81/41	−243/122

二、切线刚度法

有限元系统的切线刚度矩阵为

$$\boldsymbol{K}_T=\frac{EA}{l}\begin{bmatrix} E_T/E & -E_T/E & \\ -E_T/E & E_T/E+1 & -1 \\ & -1 & 1 \end{bmatrix}$$

式中，E_T 为单元①的切线拉伸模量，

$$E_T=\begin{cases} E & \varepsilon\le\sigma_s/E \\ 0 & \varepsilon>\sigma_s/E \end{cases}$$

引入约束条件后的切线刚度法迭代方程为

$$(1+\frac{E_T^{(k)}}{E})\Delta\bar{u}_2^{(k)}=3-\bar{\sigma}_1^{(k)}+\bar{\sigma}_2^{(k)}$$

$$\bar{u}_2^{(k+1)}=\bar{u}_2^{(k)}+\Delta\bar{u}_2^{(k)}\qquad\qquad \bar{u}_2^{(0)}=0$$

迭代计算结果列于表 9.2 中，迭代过程中应力的变化如图 9.9 所示。

表 9.2 切线刚度法迭代过程

k	$\bar{u}_2^{(k+1)}$	$\bar{\sigma}_1^{(k)}$	$\bar{\sigma}_2^{(k)}$	$E_T^{(k)}/E$	$\Delta\bar{u}_2^{(k)}$
0	0	0	0	1	3/2
1	3/2	1	-3/2	0	1/2
2	2	1	-2	0	0

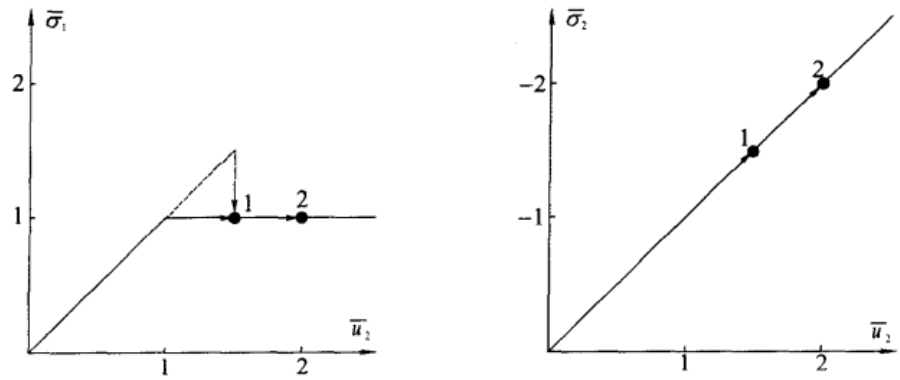


图 9.9 切线刚度法单元应力变化

三、初应力法

有限元系统的初始切线刚度矩阵为

$$\boldsymbol{K}_T^{(0)}=\frac{EA}{l}\begin{bmatrix}1 & -1 & \\ -1 & 2 & \\ & & 1\end{bmatrix}$$

引入约束条件后初应力法的迭代方程为

$$\bar{u}_2^{(k+1)}=\frac{1}{2}[3+(\bar{\sigma}_1^{e(k)}-\bar{\sigma}_1^{(k)})-(\bar{\sigma}_2^{e(k)}-\bar{\sigma}_2^{(k)})]$$

$$\bar{u}_2^{(0)}=0$$

式中，

$$\bar{\sigma}_1^{e(k)}=\sigma_1^{e(k)}/\sigma_s=E\varepsilon_1^{(k)}/\sigma_s$$

$$\bar{\sigma}_2^{e(k)}=\sigma_2^{e(k)}/\sigma_s=E\varepsilon_2^{(k)}/\sigma_s$$

迭代计算结果列于表 9.3 中，迭代过程中应力的变化如图 9.10 所示。

表 9.3 初应力法迭代过程

k	$\bar{u}_2^{(k+1)}$	$\bar{\sigma}_1^{(k)}$	$\bar{\sigma}_2^{(k)}$	$\bar{\sigma}_1^{e(k)}$	$\bar{\sigma}_2^{e(k)}$	$\Delta\bar{u}_2^{(k)}$
0	0.000	0	0.000	0.000	0.000	1.500
1	1.500	1	−1.500	1.500	−1.500	0.250
2	1.750	1	−1.750	1.750	−1.750	0.125
3	1.875	1	−1.875	1.875	−1.875	0.063
4	1.938	1	−1.938	1.938	−1.938	0.031
5	1.969	1	−1.969	1.969	−1.969	0.016
6	1.984	1	−1.984	1.984	−1.984	0.008
7	1.992	1	−1.992	1.992	−1.992	0.004
8	1.996	1	−1.996	1.996	−1.996	0.002
9	1.998	1	−1.998	1.998	−1.998	

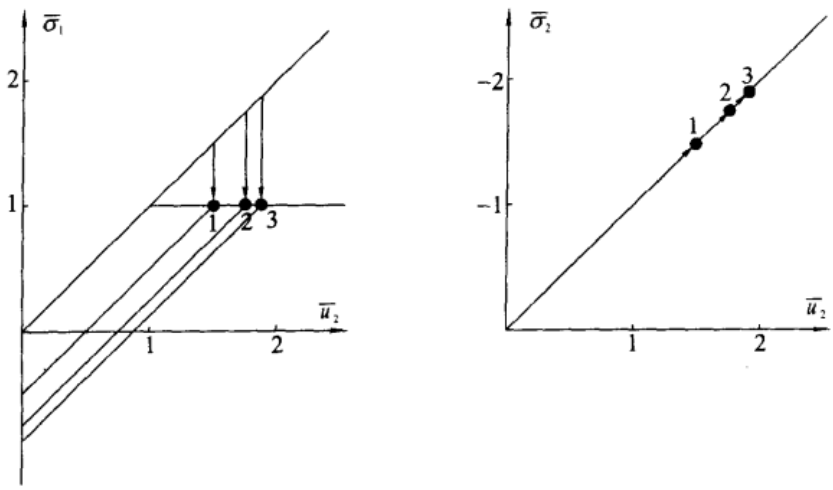


图 9.10　初应力法单元应力变化

9.2 小变形弹塑性问题的有限单元法

在弹塑性问题中，由于材料和结构的弹塑性行为与加载以及变形历史有关，材料的本构关系必须用增量形式表示。在进行结构的弹塑性有限元分析时，通常将载荷分成若干载荷增量。按比例施加外载的情况下，可以把结构不发生塑性变形的最大荷载作为第一增量，其余荷载再分成若干等份。如果实际荷载不是按比例施加的，则需要根据实际的加载次序设计载荷增量。对每一个载荷增量，将弹塑性方程线性化，从而使弹塑性分析这一非线性问题分解为一系列线性问题求解。

9.2.1 增量有限元格式

将求解的时间域划分成 n 份，记时间分点为 $0=t_0<t_1<\cdots<t_n=T$，每一个时刻 t_k 对应一定的荷载水平，即时间和载荷存在对应关系。$t=t_0$ 时刻有限元系统的位移、应变、应力、塑性应变以及其他内时变量已知。现考虑一个典型的时间步 $[t_k,t_{k+1}]$，在该时间步内，载荷增量的等效节点力为

$$\Delta \boldsymbol{F}^k=\boldsymbol{F}^{k+1}-\boldsymbol{F}^k \tag{9.26}$$

式中，$\boldsymbol{F}^k$ 和 $\boldsymbol{F}^{k+1}$ 分别为 $t=t_k$ 和 t_{k+1} 时刻的外荷载等效节点力。假设 $t_i\ (0\le i\le k)$ 时刻结构的位移、应变、应力、塑性应变以及其他内时变量均已知，分别用 $\boldsymbol{u}^i$、$\boldsymbol{\varepsilon}^i$、$\boldsymbol{\sigma}^i$、$\boldsymbol{\varepsilon}^{pi}$ 和 $\boldsymbol{\kappa}^i$ 表示。当时间由 t_k 过渡到 t_{k+1} 时，由于载荷增量 $\Delta\boldsymbol{F}^k$ 的作用，位移、应变、应力和内时变量有 $\Delta\boldsymbol{u}^k$、$\Delta\boldsymbol{\varepsilon}^k$、$\Delta\boldsymbol{\sigma}^k$、$\Delta\boldsymbol{\varepsilon}^{pk}$ 和 $\Delta\boldsymbol{\kappa}^k$ 的变化，即

$$\boldsymbol{u}^{k+1}=\boldsymbol{u}^k+\Delta\boldsymbol{u}^k \tag{9.27}$$

$$\boldsymbol{\varepsilon}^{k+1}=\boldsymbol{\varepsilon}^k+\Delta\boldsymbol{\varepsilon}^k \tag{9.28}$$

$$\boldsymbol{\varepsilon}^{pk+1} = \boldsymbol{\varepsilon}^{pk} + \Delta\boldsymbol{\varepsilon}^{pk} \tag{9.29}$$

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^{k} + \Delta\boldsymbol{\sigma}^{k} \tag{9.30}$$

$$\boldsymbol{\kappa}^{k+1} = \boldsymbol{\kappa}^{k} + \Delta\boldsymbol{\kappa}^{k} \tag{9.31}$$

在小变形条件下，应变增量和位移增量满足如下线性关系，

$$\Delta\boldsymbol{\varepsilon}^{k} = \boldsymbol{L}^{T}\Delta\boldsymbol{u}^{k} = \boldsymbol{B}\Delta\boldsymbol{d}^{k} \tag{9.32}$$

式中，用$\boldsymbol{d}^{k}$代表t_k时刻有限元系统的节点位移矢量，$\Delta\boldsymbol{d}^{k}$代表节点位移增量。t_{k+1}时刻的应力应满足如下形式的平衡方程，

$$\int_V \boldsymbol{B}^{T}\boldsymbol{\sigma}^{k+1}\mathrm{d}V = \boldsymbol{F}^{k+1} \tag{9.33}$$

将(9.30)式代入上式，得

$$\int_V \boldsymbol{B}^{T}\Delta\boldsymbol{\sigma}^{k}\mathrm{d}V = \boldsymbol{F}^{k+1} - \int_V \boldsymbol{B}^{T}\boldsymbol{\sigma}^{k}\mathrm{d}V \tag{9.34}$$

如果$\boldsymbol{\sigma}^{k}$是t_k时刻有限元问题的精确解，则有

$$\int_V \boldsymbol{B}^{T}\boldsymbol{\sigma}^{k}\mathrm{d}V = \boldsymbol{F}^{k} \tag{9.35}$$

于是(9.34)式可以改写为

$$\int_V \boldsymbol{B}^{T}\Delta\boldsymbol{\sigma}^{k}\mathrm{d}V = \Delta\boldsymbol{F}^{k} \tag{9.36}$$

但在实际计算中，我们不可能得到t_k时刻有限元问题的精确解，为避免误差累计，一般仍采用(9.34)式求解材料非线性问题。

表面上看，在小变形条件下式(9.34)是关于应力增量的线性方程组，但实际上它并不能直接求解。如果将应力增量用节点位移增量表示出来，式(9.34)则成为关于位移增量$\Delta\boldsymbol{d}^{k}$的非线性方程组。求解这个方程组就可以得到有限元系统的位移增量。而要求解该方程，我们首先必须知道应力增量和应变增量间的关系式。第 5 章的弹塑性本构理论告诉我们，弹塑性本构关系是以无限小应力增量和无限小应变增量的形式给出的，即

$$\mathrm{d}\boldsymbol{\sigma} = \boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon} = (\boldsymbol{D} - \boldsymbol{D}_{p})\mathrm{d}\boldsymbol{\varepsilon} \tag{9.37}$$

因此，$\Delta\boldsymbol{\sigma}^{k}$必须通过积分下列非线性关系得到

$$\Delta\boldsymbol{\sigma}^{k} = \int_0^{\Delta\boldsymbol{\varepsilon}^{k}} \boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon} = \int_{t_k}^{t_{k+1}} \boldsymbol{D}_{ep}\frac{\partial\boldsymbol{\varepsilon}}{\partial\tau}\mathrm{d}\tau \tag{9.38}$$

上式可以转化为一个初值问题求解。一种最简单的求解方法就是取$\boldsymbol{D}_{ep} = \boldsymbol{D}_{ep}^{k}$，它相当于采用显式欧拉法一步求解(9.38)式，在有限元分析中称为起点切线刚度法。当然在实际分析中，我们也可以采用其他方法(如预估-校正法、龙格-库塔法等)以提高应力的计算精度，但方法的选取还应兼顾计算效率。

9.2.2 非线性方程组的求解

现在我们来讨论非线性方程组(9.34)式的求解方法。根据具体问题的特点，常见的有以下几种方法可供选择。

一、无迭代增量法

无迭代增量法假设从t_k到t_{k+1}时刻弹塑性矩阵变化不大，满足

$$\boldsymbol{D}_{ep} \approx \boldsymbol{D}_{ep}^{k} \tag{9.39}$$

从而将有限应力增量和有限应变增量关系线性化。将上式和(9.32)代入(9.38)式，然后代入(9.34)式，有

$$\boldsymbol{K}_T^k \Delta \boldsymbol{d}^k = \boldsymbol{F}^{k+1} - \int_V \boldsymbol{B}^T \boldsymbol{\sigma}^k \mathrm{d}V = \boldsymbol{R}^{k+1} \tag{9.40}$$

式中，

$$\boldsymbol{K}_T^k = \int_V \boldsymbol{B}^T \boldsymbol{D}_{ep}^k \boldsymbol{B} \mathrm{d}V \tag{9.41}$$

为 t_k 时刻单元的切线刚度矩阵。

由于应力应变关系的线性化，(9.40)式的解并不能使式(9.34)精确成立。为保证有限元解具有足够的精确度，必须采用足够小的载荷增量。因此，势必增加计算载荷步，导致舍入误差的累计。同时，该方法每一个增量步都要重新生成和分解刚度矩阵，计算效率也不高。对一般的静力弹塑性问题，不建议采用这一方法。

二、具有 Newton 迭代的增量法

这一求解方法就是将第 2 章介绍的 Newton-Raphson 迭代法和载荷增量法相结合求解弹塑性问题，即在每一个载荷步中采用 Newton-Raphson 法进行迭代，直到收敛为止。为此，将(9.34)式改写为如下迭代格式，

$$\int_V \boldsymbol{B}^T \Delta \boldsymbol{\sigma}^{k(i)} \mathrm{d}V = \boldsymbol{F}^{k+1} - \int_V \boldsymbol{B}^T \boldsymbol{\sigma}^{k+1(i)} \mathrm{d}V \tag{9.42}$$

式中，上标 $i = 0,1,2,\cdots$ 代表迭代步，$\Delta \boldsymbol{\sigma}^{k(i)}$ 代表第 $k+1$ 个载荷步第 i 个迭代步的应力增量。

$$\Delta \boldsymbol{\sigma}^k = \sum_i \Delta \boldsymbol{\sigma}^{k(i)} \tag{9.43}$$

$$\boldsymbol{\sigma}^{k+1(i+1)} = \boldsymbol{\sigma}^{k+1(i)} + \Delta \boldsymbol{\sigma}^{k+1(i)} = \boldsymbol{\sigma}^k + \sum_{j=0}^{i} \Delta \boldsymbol{\sigma}^{k(j)} \tag{9.44}$$

将(9.42)式左端线性化，即令

$$\Delta \boldsymbol{\sigma}^{k(i)} = \boldsymbol{D}_T^{k+1(i)} \Delta \boldsymbol{\varepsilon}^{k(i)} = \boldsymbol{D}_T^{k+1(i)} \boldsymbol{B} \Delta \boldsymbol{d}^{k(i)} \tag{9.45}$$

则有

$$\boldsymbol{K}_T^{k+1(i)} \Delta \boldsymbol{d}^{k(i)} = \boldsymbol{R}^{k+1(i)} \tag{9.46}$$

式中，

$$\boldsymbol{K}_T^{k+1(i)} = \int_V \boldsymbol{B}^T \boldsymbol{D}_T^{k+1(i)} \boldsymbol{B} \mathrm{d}V \tag{9.47}$$

$$\boldsymbol{R}^{k+1(i)} = \boldsymbol{F}^{k+1} - \int_V \boldsymbol{B}^T \boldsymbol{\sigma}^{k+1(i)} \mathrm{d}V \tag{9.48}$$

具体迭代步骤如下，

(1) 令 $k=0$，$\boldsymbol{d}^k = \boldsymbol{0}$；

(2) 令 $i=0$，$\boldsymbol{d}^{k+1(0)} = \boldsymbol{d}^k$；

(3) 计算 $\boldsymbol{K}_T^{k+1(i)}$ 和 $\boldsymbol{R}^{k+1(i)}$；

(4) 求解迭代步位移增量 $\Delta \boldsymbol{d}^{k(i)} = (\boldsymbol{K}_T^{k+1(i)})^{-1} \boldsymbol{R}^{k+1(i)}$；

(5) 计算迭代步位移 $\boldsymbol{d}^{k+1(i+1)} = \boldsymbol{d}^{k+1(i)} + \Delta \boldsymbol{d}^{k(i)}$；

(6) 判断迭代是否收敛，如果收敛，令 $k = k+1$，转步骤(2)进行下一增量步分析；否则，

令 $i=i+1$，转步骤(3)进行下一迭代步分析。

三、具有修正 Newton 迭代的增量法

求解步骤和前一种方法相同，只是切线刚度矩阵始终保持某时刻的数值。如保持为每个载荷增量步开始时刻的数值，则称为起点切线刚度法，此时

$$\boldsymbol{K}_T^{k+1(i)}=\boldsymbol{K}_T^{k+1(0)}=\int_V \boldsymbol{B}^T\boldsymbol{D}_T^k\boldsymbol{B}\mathrm{d}V \tag{9.49}$$

为提高求解效率，当前载荷增量步内的切线刚度矩阵并不局限为起点的切线刚度，可以规定经过一定迭代次数仍不满足收敛准则的要求时，根据当前迭代步的计算结果重新生成和分解新的切线刚度矩阵。如果迭代始终采用 $t=t_0$ 时刻的弹性刚度矩阵，其解法等同于一般的初应力法，此时，

$$\boldsymbol{K}_T^{k+1(i)}=\boldsymbol{K}_T^0=\int_V \boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}\mathrm{d}V \tag{9.50}$$

这一方法的优点是整个求解过程中切线刚度矩阵只需形成和分解一次，其缺点是收敛速度比较慢，有时甚至不收敛。

以上介绍了三种常用的方法，各有各的优缺点，具体采用何种求解方法应根据具体问题的特点，综合考虑计算精度和计算效率两个方面的因素。其中第二种求解方法具有良好的收敛性，允许采用较大的载荷步。其缺点是每次迭代都要重新形成和分解新的刚度矩阵。第三种方法可以节省反复形成和分解刚度矩阵的计算量，但收敛速度要慢一些，特别是在接近荷载极限状况时。这时往往需要采用其他加速迭代措施。同时，求解方法的选择还受制于具体问题的其他一些因素，如蠕变问题、固结问题和瞬态响应问题的时间因素等。

单元的弹塑性刚度矩阵的数值积分是所有求解方法都不可回避的。由于弹塑性问题的特殊性，同一时刻单元不同位置的 $\boldsymbol{D}_T$ 可能是不同的，因此提高了被积函数的阶次。为保证数值积分的精度，数值积分的阶次应作相应的提高。通常弹塑性刚度矩阵的积分阶次要比弹性刚度矩阵高 1～2 阶。另外，最大值应力一般出现在结构的表面（也是单元的边界），因此，总是单元的边界先屈服，进入塑性状态。而 Gauss 积分方法没有在积分边界上布置求积节点，不能及时准确地判断单元开始进入塑性的时刻，从而影响计算的精度。当必须考虑此因素时，可以采用第 2 章介绍的 Labatto 积分方法。

9.2.3 弹塑性状态的确定

从求解方法可以看出，线性化后的方程组的系数矩阵和右端项的确定都是基于已经求得上一增量步或迭代步结束时的位移、应变和应力的。因此，为了进行下一增量步或迭代步的计算，需要根据当前载荷步或迭代步计算得到的位移增量来决定应变增量、应力增量和内时变量的变化，从而得到当步结束时的弹塑性状态。这一步骤称为状态确定，它和状态方程的建立以及方程组的求解一起构成了弹塑性问题非线性分析的基本算法过程，因此具有十分重要的意义。

弹塑性状态确定的一般算法步骤如下。

(1) 根据当前增量步或迭代步的位移增量 $\Delta\boldsymbol{d}^k$ 计算应变增量 $\Delta\boldsymbol{\varepsilon}^k=\boldsymbol{B}\Delta\boldsymbol{d}^k$。

(2) 按弹性应力应变关系计算应变增量对应的弹性应力增量 $\Delta\boldsymbol{\sigma}^{ek}=\boldsymbol{D}\Delta\boldsymbol{\varepsilon}^k$ 和试应力 $\boldsymbol{\sigma}^{tk+1}=\boldsymbol{\sigma}^k+\Delta\boldsymbol{\sigma}^{ek}$。

(3) 计算屈服函数值 $f(\boldsymbol{\sigma}^{tk+1},\boldsymbol{\sigma}^{pk},\kappa^k)$，式中，$\boldsymbol{\sigma}^{pk},\kappa^k$ 分别为上一载荷步结束时的塑性应力矢量和内时变量。根据计算值分三种情况：

(i) 若 $f(\boldsymbol{\sigma}^{tk+1},\boldsymbol{\sigma}^{pk},\kappa^k)\le 0$，则该点为弹性变形，满足线性弹性应力应变本构关系，试应力就是当前载荷步或迭代步的真实应力。为统一起见，令 $m=1$。

(ii) 若 $f(\boldsymbol{\sigma}^{tk+1},\boldsymbol{\sigma}^{pk},\kappa^k)>0$，且 $f(\boldsymbol{\sigma}^{k},\boldsymbol{\sigma}^{pk},\kappa^k)<0$，则该点为由弹性进入塑性的过渡过程。为简化起见，假设载荷增量过程中应变增量成比例变化，由

$$f(\boldsymbol{\sigma}^{k}+m\Delta\boldsymbol{\sigma}^{ek},\boldsymbol{\sigma}^{pk},\kappa^k)=0 \tag{9.51}$$

计算比例因子 m。根据 m 可以确定应力点到达屈服面的时刻。注意由(9.51)式求得的解可能不止一个，取 0 到 1 之间的解。

(iii) 若 $f(\boldsymbol{\sigma}^{tk+1},\boldsymbol{\sigma}^{pk},\kappa^k)>0$，且 $f(\boldsymbol{\sigma}^{k},\boldsymbol{\sigma}^{pk},\kappa^k)=0$，则该点为塑性加载，取 $m=0$。

(4) 根据 m 将应变分成弹性和弹塑性两个部分，对应弹性应变 $m\Delta\boldsymbol{\varepsilon}^k$ 的应力增量为

$$\Delta\boldsymbol{\sigma}'^{k}=m\boldsymbol{D}\Delta\boldsymbol{\varepsilon}^k \tag{9.52}$$

对应弹塑性应变 $(1-m)\Delta\boldsymbol{\varepsilon}^k$ 的应力增量为

$$\Delta\boldsymbol{\sigma}''^{k}=\int_0^{(1-m)\Delta\boldsymbol{\varepsilon}^k}\boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon} \tag{9.53}$$

计算上式时可同时获得内时变量的增量，具体的求解方法将在下面介绍。

(5) 当步的应力计算值为

$$\boldsymbol{\sigma}^{k+1}=\boldsymbol{\sigma}^{k}+\Delta\boldsymbol{\sigma}'^{k}+\Delta\boldsymbol{\sigma}''^{k} \tag{9.54}$$

要强调的是，以上状态确定过程中采用的试应力，即是弹性的应力增量，而不是应力增量(在进行判断的时候它还没有计算出来)。可以证明，上述过程实质上采用了应变空间表述的加-卸载准则。

9.2.4　本构关系积分

本构关系的积分式(9.53)的求解也是弹塑性有限单元分析的关键技术之一。针对一些特殊情况，积分式已获得解析解或解析的近似解。如 Key 得到了理想弹塑性本构关系的解析解，王勖成给出了运动强化材料本构关系的解析解和各向同性强化材料本构关系的以 $E^p/(3G+E^p)$ 为小参数的渐进解。这些解析解使得利用较小的工作量获得较精确的结果成为现实。但由于本构关系的复杂多样性，大部分程序仍然采用数值积分方法求解(9.53)式。以下介绍目前应用最多的切向预测径向返回的子增量法。

一、切向预测径向返回

所谓切向预测就是将显式欧拉法应用于(9.53)式，得到应力增量的预测值，

$$\Delta\tilde{\boldsymbol{\sigma}}''^{k}=(1-m)\boldsymbol{D}_{ep}(\boldsymbol{\sigma}^{k}+\Delta\boldsymbol{\sigma}'^{k},\boldsymbol{\sigma}^{pk},\kappa^k)\Delta\boldsymbol{\varepsilon}^k \tag{9.55}$$

应力的预测值则为

$$\tilde{\boldsymbol{\sigma}}^{k+1}=\boldsymbol{\sigma}^{k}+\Delta\boldsymbol{\sigma}'^{k}+\Delta\tilde{\boldsymbol{\sigma}}''^{k} \tag{9.56}$$

将(5.51)式中的 $\mathrm{d}\lambda$ 和 $\mathrm{d}\boldsymbol{\varepsilon}$ 改写成有限增量形式，得

$$\Delta\lambda^k=\frac{1-m}{A^k}\left(\frac{\partial f^k}{\partial\boldsymbol{\sigma}}\right)^T\boldsymbol{D}\Delta\boldsymbol{\varepsilon}^k \tag{9.57}$$

$$A^k=(\frac{\partial f^k}{\partial \boldsymbol{\sigma}})^T \boldsymbol{D}\frac{\partial f^k}{\partial \boldsymbol{\sigma}}\frac{\partial f^k}{\partial \boldsymbol{\sigma}^p}\boldsymbol{D}\frac{\partial f^k}{\partial \boldsymbol{\sigma}}-M^k \tag{9.58}$$

$$M^k=\begin{cases}\dfrac{\partial f^k}{\partial \kappa}(\boldsymbol{\sigma}^k+\Delta\boldsymbol{\sigma}'^k)^T\dfrac{\partial f^k}{\partial \boldsymbol{\sigma}} & \text{当}\kappa=w^p\text{时}\\ \dfrac{\partial f^k}{\partial \kappa}\boldsymbol{M}^T\dfrac{\partial f^k}{\partial \boldsymbol{\sigma}} & \text{当}\kappa=\theta^p\text{时}\\ \dfrac{\partial f^k}{\partial \kappa}\left[(\dfrac{\partial f^k}{\partial \boldsymbol{\sigma}})^T\dfrac{\partial f^k}{\partial \boldsymbol{\sigma}}\right]^{\frac{1}{2}} & \text{当}\kappa=\bar{\varepsilon}^p\text{时}\end{cases} \tag{9.59}$$

式中，

$$\frac{\partial f^k}{\partial \boldsymbol{\sigma}}=\frac{\partial f(\boldsymbol{\sigma}^k+\Delta\boldsymbol{\sigma}'^k,\boldsymbol{\sigma}^{pk},\kappa^k)}{\partial \boldsymbol{\sigma}}$$

$$\frac{\partial f^k}{\partial \kappa}=\frac{\partial f(\boldsymbol{\sigma}^k+\Delta\boldsymbol{\sigma}'^k,\boldsymbol{\sigma}^{pk},\kappa^k)}{\partial \kappa}$$

由(5.46)和(5.58)式可得

$$\Delta\boldsymbol{\varepsilon}^{pk}=\Delta\lambda^k\frac{\partial f^k}{\partial \boldsymbol{\sigma}} \tag{9.60}$$

其他的内时变量，如$\Delta\kappa^k$，可以同样求得。

由于(9.55)式采用了增量$(1-m)\Delta\boldsymbol{\varepsilon}^k$起点时刻的切线弹性矩阵，同时注意到屈服面是外凸的，所以$\Delta\tilde{\boldsymbol{\sigma}}''^k$是在屈服面的切线方向，且$\tilde{\boldsymbol{\sigma}}^{k+1}$总在屈服面的外侧。但是屈服准则要求$t_{k+1}$时刻的应力只能在屈服面上，为此采用径向返回的方法以求得满足屈服条件的应力$\boldsymbol{\sigma}^{k+1}$。

$$f(\gamma\tilde{\boldsymbol{\sigma}}^{k+1},\gamma\tilde{\boldsymbol{\sigma}}^{pk+1},\kappa^{k+1})=0 \tag{9.61}$$

式中，$\tilde{\boldsymbol{\sigma}}^{pk+1}$为切向预测的应力。由上式求得$\gamma$后得到$t_{k+1}$时刻的应力，

$$\boldsymbol{\sigma}^{k+1}=\gamma\tilde{\boldsymbol{\sigma}}^{k+1} \tag{9.62}$$

必须注意的是，经过上式校正的应力虽然在屈服面上，但它仍然不是真实的应力，因为(9.61)式假定塑性应变增量保持不变。当应变步长较大时，这种区别可能是不可忽略的。为减少误差，通常采用子增量法。

二、子增量法

顾名思义，子增量法就是将总的弹塑性应变增量分成若干个子增量。对于每个子增量，利用上述切向预测径向返回的方法确定应力增量，将每一个子增量结束时的弹塑性状态作为下一个子增量的初始状态。子增量有助于提高应力增量的计算精度，从而加快迭代的收敛速度。当然，并不是子增量步的数目越多越好，因为子增量数目的增加使得计算量也相应增加。因此，合理的确定子增量的数目是非常重要的。下面介绍三种常用的确定方法：

(1) 根据弹性应变偏量增量的大小$\Delta\bar{e}^{ek}$确定子增量数N，即

$$N=1+\frac{\Delta\bar{e}^{ek}}{\Delta\bar{e}_0} \tag{9.63}$$

式中，

$$\Delta\bar{e}^{ek}=(\frac{2}{3}\Delta e_{ij}^{ek}e_{ij}^{ek})^{\frac{1}{2}}$$

$$\Delta e_{ij}^{ek}=(1-m)\Delta e_{ij}^{k}-\Delta e_{ij}^{pk}$$

$\Delta\bar{e}_0$ 应根据计算精度的要求加以选择。计算经验认为取 $\Delta\bar{e}_0=0.0002$ 就可以得到满意的结果。实际计算时，建议综合考虑计算精度和初始屈服时对应的应变两个因素。

(2) 根据应力偏量 $s_{ij}^k+\Delta s_{ij}'^k$ 和 $\tilde{s}_{ij}^{k+1}$ 之间的夹角确定子增量数 N，即

$$N=1+\frac{\psi^k}{\psi_0} \tag{9.64}$$

其中，

$$\psi=\arccos(\frac{(s_{ij}^k+\Delta s_{ij}'^k)\tilde{s}_{ij}^{k+1}}{\sqrt{(s_{ij}^k+\Delta s_{ij}'^k)(s_{ij}^k+\Delta s_{ij}'^k)\tilde{s}_{ij}^{k+1}\tilde{s}_{ij}^{k+1}}})$$

式中，ψ_0 根据计算精度要求确定。

(3) 按等效应力确定子增量数 N，即

$$N=8\times\frac{\tilde{\sigma}_y-\sigma_y'}{\sigma_y^0}+1 \tag{9.65}$$

式中，σ_y^0 为初始屈服时对应的等效应力，$\tilde{\sigma}_y$ 为按试应力计算得到的等效应力，σ_y' 为 $\boldsymbol{\sigma}^k+\Delta\boldsymbol{\sigma}'^k$ 对应的等效应力。

9.3　突加载荷作用下理想弹塑性梁的非线性振动

本节对矩形脉冲载荷作用下理想弹塑性梁的振动进行非线性分析，研究其动力分叉现象。如图9.11所示连接在两个光滑固定铰间的梁，承受均匀矩形脉冲载荷 p 的作用。P.S.Symonds和T.X.Yu[1]报道了一种反直觉现象：在特定的载荷(如 $p_0=19.2$ N/mm)作用下，10种不同的程序得到相去很远的结果，而且，梁在经过和载荷同向的最大位移后，可能反向振动至与载荷相反的方向，并在该方向的某一位置附近作弹性振动。S.H.Lee和S.S.Hsieh[2]指出，在一些特定值附近，载荷的微小变化会得到相差很大的响应，并认为这是一种动力分叉现象。

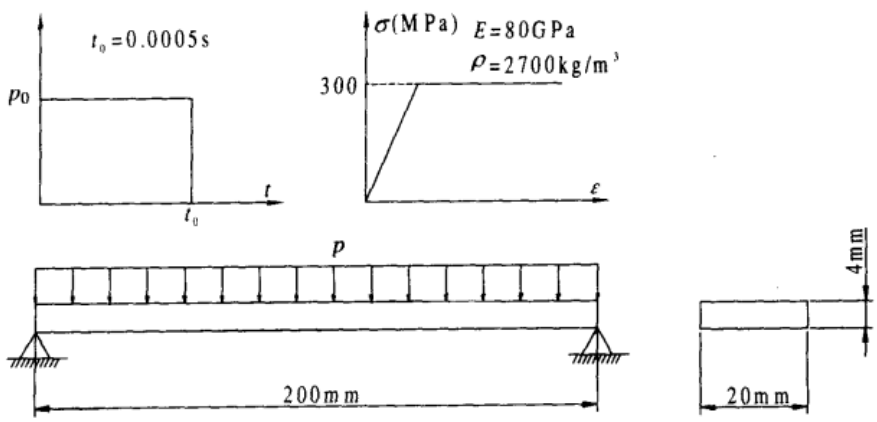


图9.11　突加载荷作用下的弹塑性单层梁

9.3.1 弹塑性应力应变关系

假定梁为理想弹塑性材料，服从Mises屈服准则，

$$f(\boldsymbol{\sigma},\boldsymbol{\sigma}^{p},\kappa)=J_2^{1/2}-\mathrm{K}_0 \tag{9.66}$$

对于理想弹塑性材料，K_0为常数。对于梁问题，只有两个应力分量非零，此时，应力矢量、应变矢量和弹性系数矩阵为

$$\boldsymbol{\sigma}=\{\sigma,\tau\}^T \tag{9.67}$$

$$\boldsymbol{\varepsilon}=\{\varepsilon,\gamma\}^T \tag{9.68}$$

$$\boldsymbol{D}=\begin{bmatrix}E & 0\\ 0 & G\end{bmatrix} \tag{9.69}$$

式中，σ和τ分别为梁内轴向正应力和横向剪应力；ε和γ分别为梁内轴向正应变和横向剪应变。容易导出

$$J_2=\frac{1}{3}\sigma^2+\tau^2 \tag{9.70}$$

$$\frac{\partial f}{\partial\boldsymbol{\sigma}}=\frac{1}{\mathrm{K}_0}\{\frac{\sigma}{3},\tau\}^T \tag{9.71}$$

将以上各式代入(5.59)~(5.61)式，得

$$\boldsymbol{D}_p=\frac{1}{E\sigma^2+9G\tau^2}\begin{bmatrix}E^2\sigma^2 & 3EG\sigma\tau\\ 3EG\sigma\tau & 9G^2\tau^2\end{bmatrix} \tag{9.72}$$

由于材料非线性的引入，沿厚度方向单元刚度矩阵不可能精确积分，必须借助于数值积分法。为保证积分精度，沿轴线方向采用高阶高斯积分，算例中均采用五点高斯积分。沿厚度方向，由于表面首先进入屈服，而高斯积分没有在边界上的积分点，因此采用高斯积分方法将不能准确地判断单元开始进入塑性的时刻，从而影响计算精度，以下分析中采用了八点Labatto积分。

9.3.2 理想弹塑性固定简支梁的响应

材料参数和截面几何参数如图9.11所示的梁，初始时刻（$t=0$）结构处于静止状态，初位移和初速度均为零。取对称的一半进行分析，采用十个等长度梁单元，并考虑小应变中等转动几何非线性。为便于比较所有的计算采用统一的时间步长(1×10^{-5} s)和计算响应的时间长度(1.2×10^{-2} s)。

不同p_0作用下中点挠度响应如图9.12所示。计算的应变表明，在突加的均布载荷作用p（p_0=16.20~16.38N/mm）作用下，中点挠度由零逐渐增加，到5.6×10^{-4}sec左右达到峰值。挠度达到峰值之前几乎没有塑性应变产生。挠度达到峰值后迅速减少，由于纵向振动的影响，中点在平衡位置附近(挠度为-2～2mm)有较大的塑性应变产生。当p_0在16.36N/mm附近时，微小的载荷变化可能导致相差很远的响应，甚至是本质不同的。由图可见，当 p_0=16.34N/mm

时，响应呈现文献[1]所述的反直觉行为。更为有意义的是，当均布荷载在16.359828N/mm和16.359830N/mm之间会出现一个高频低幅的振动状态，显然该状态是极不稳定的，微小的干扰将会导致响应偏离该状态。

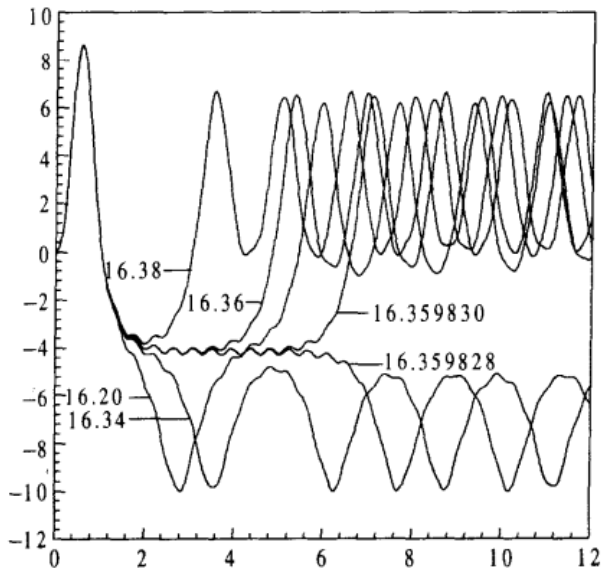


图9.12　梁中点挠度（mm）-时间（msec）曲线

9.3.3 理想弹塑性固支夹层梁的响应

夹层梁在结构上远比一般梁复杂，假定夹层梁的两表层厚度相同，且均为理想弹塑性材料，芯层为线弹性材料。夹层梁两端固支，长100mm，宽20mm，表层厚0.2mm，芯层厚3.6mm。夹层梁承受与图9.11所示梁相同的均布载荷。材料的基本力学参数如下，

表层：

$$E_f = 2.1\times10^5\,\mathrm{N/mm^2} \qquad G_f = 5.0\times10^4\,\mathrm{N/mm^2}$$

$$\sigma_s = 3.0\times10^3\,\mathrm{N/mm^2} \qquad \rho_f = 8.0\times10^{-5}\,\mathrm{N/mm^3}$$

芯层：

$$E_c = 1.6\times10^3\,\mathrm{N/mm^2} \qquad G_c = 6.4\times10^2\,\mathrm{N/mm^2}$$

$$\rho_c = 1.0\times10^{-5}\,\mathrm{N/mm^3}$$

图9.13为不同p_0作用下中点挠度的时间历程曲线。由图容易看出，在p_0的一个很小的范围内出现三次不完全相同的动力分叉，但没有发现类似单层梁的“反直觉”现象，这主要是由于假设弹性芯层引起的。与图9.12比较可以看出，即使夹层梁的表层具有较大的屈服极限，由于芯层厚而软，可能导致较小载荷下发生动力分叉，这一点对夹层结构的设计是十分重要的。

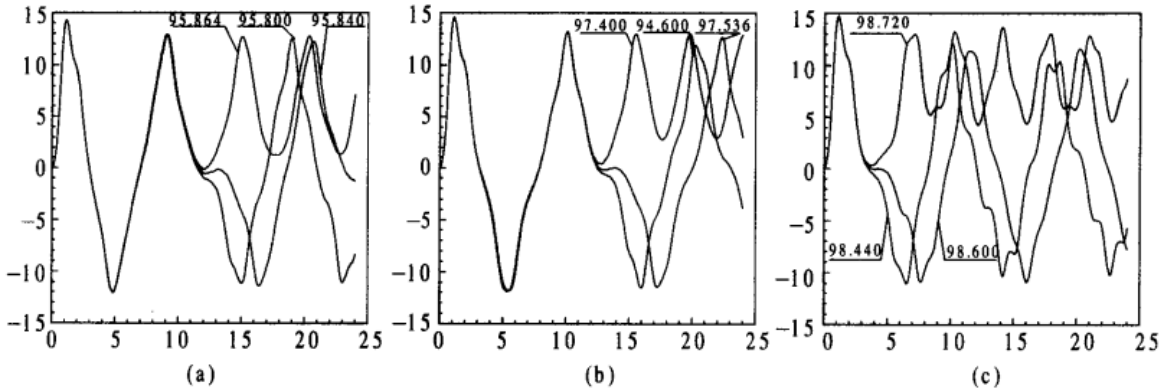


图9.13 夹层梁中点挠度（mm）-时间（sec）曲线

9.4 节理单元

在岩土工程的有限元分析中，经常会遇到节理、断层和断层带、破碎带以及其他类似土与结构连接处的软弱结构面。为模拟这类结构面的特性，需要开发一类特殊的单元，即节理单元。实际上节理单元在工程中的应用要广泛得多，譬如，节理单元可以用来模拟机械中各种配合的接触性质，这时一般称为连接单元。本节先讨论三维空间问题有限元分析中的节理单元，然后简化到二维平面问题。

9.4.1 8～20节点空间节理单元

如图9.14所示，三维空间节理单元可以看成一个厚度很小的8～20节点空间六面体等参单元，其几何形状与退化板壳单元相似，不同的只是本构关系。类似退化板壳单元，我们规定单元的1、2、3、4节点位于单元的下表面，相应的5、6、7、8节点位于单元的上表面。由于厚度很小，边中节点17、18、19、20是可以省略的，即空间节理单元一般为8～16节点。



图 9.14 空间等参元和空间节理单元

节理单元的坐标和位移插值与8～20节点六面体等参单元相同，见式(6.88)和(6.89)，位移应变关系见式(6.94)。与实体退化单元相同，以节理单元中面为基准建立局部坐标系 $o'x'y'z'$，

建立步骤参见 7.3.3 节。单元局部坐标系下的应力和应变矢量分别为

$$\boldsymbol{\sigma}' = \{\sigma_{x'}, \sigma_{y'}, \sigma_{z'}, \tau_{y'z'}, \tau_{z'x'}, \tau_{x'y'}\}^T \tag{9.73}$$

$$\boldsymbol{\varepsilon}' = \{\varepsilon_{x'}, \varepsilon_{y'}, \varepsilon_{z'}, \gamma_{y'z'}, \gamma_{z'x'}, \gamma_{x'y'}\}^T \tag{9.74}$$

下面我们首先确定节理介质的本构关系。采用层状材料的屈服函数(5.95)式，可以导出单元局部坐标系下弹塑性应力应变增量本构关系，

$$\mathrm{d}\boldsymbol{\sigma}' = (\boldsymbol{D} - H(L)\boldsymbol{D}'_p)\mathrm{d}\boldsymbol{\varepsilon}' = \boldsymbol{D}'_{ep}\mathrm{d}\boldsymbol{\varepsilon}' \tag{9.75}$$

式中，

$$\boldsymbol{D}'_p = \frac{1}{A}\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} \\ A_{61} & A_{62} & A_{63} & A_{64} & A_{65} & A_{66} \end{bmatrix}$$

其中，

$$A = \left(K + \frac{4}{3}G\right)\mu^2 + G\frac{\beta^2 - a^2c^2}{\beta^2} + \mu(c' - \mu'\sigma_{z'}) - a^2cc'\mu/\beta$$

$$A_{11} = A_{22} = A_{12} = A_{21} = \mu^2(K - \frac{2}{3}G)^2$$

$$A_{13} = A_{23} = A_{31} = A_{32} = \mu^2(K - \frac{2}{3}G)(K + \frac{4}{3}G)$$

$$A_{33} = (K + \frac{4}{3}G)^2\mu^2$$

$$A_{14} = A_{24} = A_{41} = A_{42} = \mu G(K - \frac{2}{3}G)\frac{\tau_{y'z'}}{\beta}$$

$$A_{34} = A_{43} = \mu G(K + \frac{4}{3}G)\frac{\tau_{y'z'}}{\beta}$$

$$A_{44} = G^2\frac{\tau_{y'z'}^2}{\beta^2}$$

$$A_{15} = A_{25} = A_{51} = A_{52} = \mu G(K - \frac{2}{3}G)\frac{\tau_{z'x'}}{\beta}$$

$$A_{35} = A_{53} = \mu G(K + \frac{4}{3}G)\frac{\tau_{z'x'}}{\beta}$$

$$A_{45} = A_{54} = G^2\frac{\tau_{y'z'}\tau_{z'x'}}{\beta^2}$$

$$A_{55} = G^2\frac{\tau_{z'x'}^2}{\beta^2}$$

$$A_{6i} = A_{i6} = 0 \quad i = 1, 2, \cdots, 6$$

以上各式中符号的含义见 5.5.4 节。将(9.75)式进行坐标变换，可得到整体坐标系下的应力应变

增量本构关系，

$$\mathrm{d}\boldsymbol{\sigma} = \boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon} \tag{9.76}$$

式中，

$$\boldsymbol{D}_{ep} = \boldsymbol{T}_{\varepsilon}^{T}\boldsymbol{D}'_{ep}\boldsymbol{T}_{\varepsilon} \tag{9.77}$$

$\boldsymbol{T}_{\varepsilon}$ 的表达式见(7.53)式。

知道了节理单元的本构矩阵 $\boldsymbol{D}_{ep}$ 和单元节点位移与应变关系矩阵 $\boldsymbol{B}$，就容易写出节理单元的刚度矩阵

$$\boldsymbol{K}^{e} = \int_{V'}\boldsymbol{B}^{T}\boldsymbol{D}_{ep}\boldsymbol{B}\mathrm{d}V \tag{9.78}$$

以及荷载的等效节点力矢量等。计算 $\boldsymbol{K}^{e}$ 时可以采用 Gauss 积分或 Labatto 积分，这里不再详细说明。如果假设应力和应变沿厚度方向均匀变化，式(9.78)可简化为面积积分，当然也可以在厚度方向设置一个积分点。

9.4.2 4～8 节点平面节理单元

我们可以很容易地将上述推导推广到平面应变节理问题的分析，构造平面应变节理单元。如图 9.15 所示，将平面节理单元看成一种特殊的 4～8 节点平面等参单元。单元坐标和位移插值及单元应变与节点位移的关系与平面应变等参单元完全相同，不同的只有应力应变关系。

下面我们来建立平面应变节理单元的弹塑性增量应力应变关系。如图 9.15 所示，规定边 12 和 34 分别位于节理的上下表面。以 $\eta = 0$ 为基准建立单元局部坐标系 $o'x'y'$。单元内任意点 (ξ,η) 的单位坐标方向矢量按以下步骤定义。

(1)计算矢量 $\mathbf{V}_1 = \{\bar{l}_{x'}, \bar{m}_{x'}\}^{T}$，其中，

$$\bar{l}_{x'} = \sum_{i=1}^{n}\frac{\partial N_i(\xi,0)}{\partial\xi}x_i \qquad \bar{m}_{x'} = \sum_{i=1}^{n}\frac{\partial N_i(\xi,0)}{\partial\xi}y_i \tag{9.79}$$

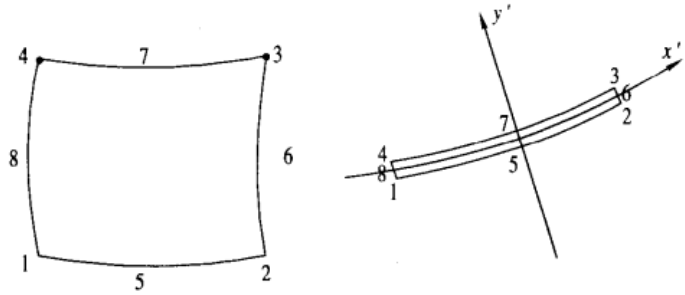


图 9.15 平面等参元和平面节理单元

(2)计算 x' 坐标线的单位切线方向矢量 $\boldsymbol{i}_{x'} = \{l_{x'}, m_{x'}\}^{T}$，其中，

$$l_{x'} = \frac{\bar{l}_{x'}}{L_{x'}} \qquad m_{x'} = \frac{\bar{m}_{x'}}{L_{x'}} \tag{9.80}$$

其中，

$$L_{x'} = \sqrt{\bar{l}_{x'}^{2} + \bar{m}_{x'}^{2}}$$

(3)计算 y' 坐标线的单位切线方向矢量 $\boldsymbol{i}_{y'}=\{l_{y'},m_{y'}\}^T$，其中，

$$l_{y'}=-m_{x'} \qquad m_{y'}=l_{x'} \tag{9.81}$$

对于平面应变问题，$\varepsilon_z=0$，$\gamma_{yz}=\gamma_{xz}=0$，因此，

$$\varepsilon_{z'}=0 \qquad \gamma_{y'z'}=\gamma_{x'z'}=0 \tag{9.82}$$

在空间节理单元的 $\boldsymbol{D}'_p$ 矩阵中引入上述条件就可以得到平面节理单元的 $\boldsymbol{D}'_p$ 矩阵

$$\boldsymbol{D}'_p=\frac{1}{A}\begin{bmatrix}A_{11} & A_{12} & A_{13}\\ A_{21} & A_{22} & A_{23}\\ A_{31} & A_{32} & A_{33}\end{bmatrix} \tag{9.83}$$

其中，

$$A=\left(K+\frac{4}{3}G\right)\mu^2+G\frac{\beta^2-a^2c^2}{\beta^2}+\mu(c'-\mu'\sigma_{y'})-a^2cc'\mu/\beta$$

$$A_{11}=\mu^2(K-\frac{2}{3}G)^2$$

$$A_{12}=A_{21}=\mu^2(K-\frac{2}{3}G)(K+\frac{4}{3}G)$$

$$A_{22}=(K+\frac{4}{3}G)^2\mu^2$$

$$A_{13}=A_{31}=\mu G(K-\frac{2}{3}G)\frac{\tau_{x'y'}}{\beta}$$

$$A_{23}=A_{32}=\mu G(K+\frac{4}{3}G)\frac{\tau_{x'y'}}{\beta}$$

$$A_{33}=G^2\frac{\tau_{x'y'}^2}{\beta^2}$$

由此不难得到平面节理单元弹塑性应力应变增量本构关系，

$$\mathrm{d}\boldsymbol{\sigma}'=(\boldsymbol{D}-H(L)\boldsymbol{D}'_p)\mathrm{d}\boldsymbol{\varepsilon}'=\boldsymbol{D}'_{ep}\mathrm{d}\boldsymbol{\varepsilon}' \tag{9.84}$$

经过坐标变换可得到整体坐标系下的弹塑性应力应变增量本构关系，

$$\mathrm{d}\boldsymbol{\sigma}=\boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon} \tag{9.85}$$

式中，

$$\boldsymbol{D}_{ep}=\boldsymbol{T}_\varepsilon^T\boldsymbol{D}'_{ep}\boldsymbol{T}_\varepsilon \tag{9.86}$$

其中，

$$\boldsymbol{T}_\varepsilon=\begin{bmatrix}l_{x'}l_{x'} & m_{x'}m_{x'} & l_{x'}m_{x'}\\ l_{y'}l_{y'} & m_{y'}m_{y'} & l_{y'}m_{y'}\\ 2l_{x'}l_{y'} & 2m_{x'}m_{y'} & l_{x'}m_{y'}+l_{y'}m_{x'}\end{bmatrix}$$

类似地，我们可以得到单元的刚度矩阵和荷载等效节点力矢量。

9.5 小变形弹塑性有限元分析的基本模块

下面给出小变形弹塑性有限元分析中常用到的一些公共模块。读者在编制单元时可以根据需要直接调用。以下所有子程序均在文件 Plastic.f90 中。

```
!=================================================================
!       |  文件功能:
!       |      本文件的各个子程序为实现弹塑性分析需要的公共模块
!       |      创建者:凌道盛
!       |      时  间:2002/04/13
!       |      变量说明:
!       |                sflow — 流动矢量(屈服函数对应力矢量的偏导数)
!       |                pflow — 塑性流动矢量(屈服函数对塑性应力矢量的偏导数)
!       |                dstrn — 应变增量
!       |                strsp — 预应力
!       |                dmate — 弹性矩阵
!       |                dmatp — 弹塑性矩阵
!       |                hardk — 强化规律(屈服函数对内时变量 κ 的偏导数)
!=================================================================
      DOUBLE PRECISION FUNCTION Lamda( dmate, strsg, strsp, dstrn,      &
                                       sflow, pflow, hardk, ietyp, itype )
!     计算 dλ
      USE CtrlData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION sflow( 6 ), pflow( 6 )
      DIMENSION vwork( 6 ), dstrn( 6 )
      DIMENSION strsp( 6 ), strsg( 6 ), dmate( 6, 6 )
!     计算(塑性)流动矢量转置与弹性矩阵的乘积
      DO istrs = 1, 6
        vwork( istrs ) = 0.0D0
        DO jstrs = 1, 6
          vwork( istrs ) = vwork( istrs ) +                             &
          sflow( jstrs ) * dmate( jstrs, istrs )
        END DO
      END DO
!     先计算 dλ 的分子部分
      partn = 0.0D0
      DO istrs = 1, 6
        partn = partn + vwork( istrs ) * dstrn( istrs )
      END DO
!     再计算 dλ 的分母部分
      partd = 0.0D0
      DO istrs = 1, 6
```

```
            partd = partd + vwork( istrs ) * sflow( istrs )
            partd = partd - vwork( istrs ) * pflow( istrs )
          END DO
          dkdla = DkDlamda( sflow, strsg, strsp, ietyp, itype )
          IF( nerrc .NE. 0 ) RETURN
          partd = partd - hardk * dkdla
          Lamda = partn / partd
          IF( Lamda .LT. 0.0D0 ) Lamda = 0.0D0
          END

          FUNCTION DkDlamda( sflow, strsg, strsp, ietyp, itype )
!=================================================================
!        | 模块功能：计算 dκ/dλ
!=================================================================
          USE CtrlData
          IMPLICIT DOUBLE PRECISION( a-h, o-z )
          DIMENSION strsg( 6 ), vwork( 6 )
          DIMENSION sflow( 6 ), vtemp( 6 )
          DIMENSION strsp( 6 ), vtran( 6 )

          CALL ReorderStress( strsg, vwork, ietyp, 0 )
          CALL ReorderStress( strsp, vtran, ietyp, 0 )
          CALL ReorderStress( sflow, vtemp, ietyp, 0 )
          DO istrs = 1, 6
            vwork( istrs ) = vwork( istrs ) + vtran( istrs )
          END DO

          reslt = 0.0D0
          SELECT CASE( itype )
            CASE( 1 )
!.....塑性功
              DO istrs = 1, 6
                reslt = reslt + vwork( istrs ) * vtemp( istrs )
              END DO
            CASE( 2 )
!.....塑性体积应变
              reslt = vtemp(1) + vtemp(2) + vtemp(3)
            CASE( 3 )
!.....塑性等效应变
              DO istrs = 1, 6
                reslt = reslt + vtemp( istrs ) * vtemp( istrs )
              END DO
              reslt = DSQRT( reslt )
            CASE DEFAULT
```

```
            nerrc = 3403546
            WRITE( 12, 2000 )
        END SELECT
        DkDlamda = reslt
2000    FORMAT( //2x, '致命错误：选择了无效的内时变量！' )
        END

        SUBROUTINE DEPMatix( dmate, dmatp, strsg, strsp, sflow,                    &
                             pflow, hardk, ietyp, itype )
!       计算弹塑性矩阵 Dep
        USE CtrlData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION dmate( 6, 6 ), strsg( 6 ), sflow( 6 )
        DIMENSION dmatp( 6, 6 ), pflow( 6 ), vwork( 6 ), strsp( 6 )
!       计算(塑性)流动矢量转置与弹性矩阵的乘积
        DO istrs = 1, 6
          vwork( istrs ) = 0.0D0
          DO jstrs = 1, 6
            vwork( istrs ) = vwork( istrs ) +                                      &
            sflow( jstrs ) * dmate( jstrs, istrs )
            dmatp( istrs, jstrs ) = dmate( istrs, jstrs )
          END DO
        END DO
!       计算 A 的分母部分
        apara = 0.0D0
        DO istrs = 1, 6
          apara = apara + vwork( istrs ) * sflow( istrs )
          apara = apara - vwork( istrs ) * pflow( istrs )
        END DO
        dkdla = DkDlamda( sflow, strsg, strsp, ietyp, itype )
        IF( nerrc .NE. 0 ) RETURN
        apara = apara - hardk * dkdla
        DO istrs = 1, 6
          DO jstrs = 1, 6
            dmatp( istrs, jstrs ) = dmatp( istrs, jstrs ) -                        &
            vwork( istrs ) * vwork( jstrs ) / apara
          END DO
        END DO
        END
```

以下以 Drucke-Prager 屈服模型为例说明类似弹塑性问题相关子程序的编制。

```
        SUBROUTINE ReadMatM423( imats, iposp )
        USE CtrlData
```

```
USE MeshData
IMPLICIT DOUBLE PRECISION( a-h, o-z )

READ( 11, 2000 ) elast, poiss, alpha, belta, reduc, imtyp
IF( reduc .LT. 0.01D0 ) reduc = 1.0D0

IF( nprnc .NE. 0 ) THEN
  WRITE( 12, 2200 )
  WRITE( 12, 2400 ) elast, poiss
END IF

IF( imtyp .EQ. 0 ) THEN
  IF( nprnc .NE. 0 ) WRITE( 12, 2600 ) alpha, belta
ELSE
  cohes = alpha / reduc
  angle = belta * DATAN( 1.0D0 ) / 45.0D0
  angle = DATAN( DTAN( angle ) / reduc )

  ROOT3 = DSQRT( 3.0D0 )
  sinph = DSIN( angle )
  cosph = DCOS( angle )
  IF( imtyp .EQ. 1 ) THEN
    IF( nprnc .NE. 0 ) WRITE( 12, 2601 ) alpha, belta, reduc
    const = 2.0D0 / ROOT3 / ( 3.0D0 - sinph )
    alpha = const * sinph
    belta = const * 3.0D0 * cohes * cosph
  ELSE IF( imtyp .EQ. 2 ) THEN
    IF( nprnc .NE. 0 ) WRITE( 12, 2602 ) alpha, belta, reduc
    const = 2.0D0 / ROOT3 / ( 3.0D0 + sinph )
    alpha = const * sinph
    belta = const * 3.0D0 * cohes * cosph
  ELSE IF( imtyp .EQ. 3 ) THEN
    IF( nprnc .NE. 0 ) WRITE( 12, 2603 ) alpha, belta, reduc
    const = 1.0D0 / ROOT3 / ( 3.0D0 + sinph * sinph )
    alpha = const * sinph
    belta = const * 3.0D0 * cohes * cosph
  ELSE IF( imtyp .EQ. 4 ) THEN
    IF( nprnc .NE. 0 ) WRITE( 12, 2604 ) alpha, belta, reduc
    const = 8.0D0 * ROOT3 * DATAN( 1.0D0 )
    const = const * ( 9.0D0 - sinph * sinph )
    const = 2.0D0 * ROOT3 / DSQRT( const )
    alpha = const * sinph
    belta = const * 3.0D0 * cohes * cosph
  ELSE IF( imtyp .EQ. 5 ) THEN
    IF( nprnc .NE. 0 ) WRITE( 12, 2605 ) alpha, belta, reduc
```

```
          alpha = sinph / 3.0D0
          belta = cohes * cosph
        ELSE
          nerrc = 5469456
          WRITE( 12, 2604 )
        END IF
      ENDIF

      props( iposp      , imats ) = elast
      props( iposp + 1, imats ) = poiss
      props( iposp + 2, imats ) = alpha
      props( iposp + 3, imats ) = belta

      IF( nincc .NE. 0 .AND. nincc .NE. 2 ) THEN
        WRITE( 12, 2800 )
        nerrc = 24239
      END IF
      nincc = 2

2000  FORMAT( 4E15.5 / E15.5, I5 )
2200  FORMAT( 2x, '材料类型：Drucker-Prager 模型' )
2400  FORMAT( 2x, '杨氏模量:', E15.5/2x, '泊 松 比:', F15.5 )
2600  FORMAT( 2x, 'alpha-值:', E15.5/2x, 'Karpa-值:', E15.5 )
2601  FORMAT( 2x, 'Mohr 外顶点外接圆' /                                    &
              2x, '粘 聚 力:', E15.5/2x, '内摩擦角:', E15.5/               &
              2x, '折减系数:', F15.5 )
2602  FORMAT( 2x, 'Mohr 内顶点外接圆' /                                    &
              2x, '粘 聚 力:', E15.5/2x, '内摩擦角:', E15.5/               &
              2x, '折减系数:', F15.5 )
2603  FORMAT( 2x, 'Mohr 内切圆' /                                          &
              2x, '粘 聚 力:', E15.5/2x, '内摩擦角:', E15.5/               &
              2x, '折减系数:', F15.5 )
2604  FORMAT( 2x, 'Mohr 等面积圆' /                                        &
              2x, '粘 聚 力:', E15.5/2x, '内摩擦角:', E15.5/               &
              2x, '折减系数:', F15.5 )
2605  FORMAT( 2x, 'Mohr 匹配圆' /                                          &
              2x, '粘 聚 力:', E15.5/2x, '内摩擦角:', E15.5/               &
              2x, '折减系数:', F15.5 )
2606  FORMAT( 2x, '致命错误： *** 没有定义的材料参数 ***' )
2800  FORMAT( 2x, '致命错误： *** 解法不兼容！ *** ' )
      END
```

```
      SUBROUTINE HistoryQuantityM423( nhstr )
      nhstr = 5
      RETURN
      END

      SUBROUTINE StrsM423( strng, dstrn, strsg, dstrs, strsp, ietyp,          &
                           ielem, ielmx, imats, iposh, iposp )
!
!=============================================================
!       iposh - 历史量的起始位置
!
!=============================================================
      USE CtrlData
      USE MeshData
      USE GlobData
      USE ExtraMesh
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION strng( 6 ), dstrn( 6 ), dmate( 6, 6 )
      DIMENSION strsg( 6 ), dstrs( 6 ), dmatx( 6, 6 )
      DIMENSION strst( 6 ), estrn( 6 ), strsp( 6 )
      DIMENSION sflow( 6 ), pflow( 6 ), dstrb( 6 )
      DOUBLE PRECISION Lamda
!.......先保存位移分量
      DO istrn = 1, 6
        dstrb( istrn ) = dstrn( istrn )
      END DO
!.......获得基本参数
      nstrs = NumOfStress( ietyp, 1 )
      alpha = props( iposp + 2, imats )
      belta = props( iposp + 3, imats )
      drght = belta
!.......计算上一时间步或增量步的应力和当前迭代步的试应力增量
      DO istrs = 1, nstrs
        ipose = iposh + istrs - 1
        IF( ielmx .LE. 0 ) THEN
          estrn( istrs ) = strng( istrs ) - histr( ipose, ielem )
        ELSE
          estrn( istrs ) = strng( istrs ) - histx( ipose, ielmx )
        END IF
      END DO
      CALL StrsM001( estrn, strsg, ietyp, imats, iposp )
      CALL StrsM001( dstrn, dstrs, ietyp, imats, iposp )
!.......计算上一增量步应力
      DO istrs = 1, 6
```

```
          strst( istrs ) = strsg( istrs ) + dstrs( istrs )
        END DO
!.....计算上一增量步应力屈服函数值和试应力屈服函数值
        CALL YieldM423( strsg, strsp, dyldp, ietyp, imats, iposp )
        CALL YieldM423( strst, strsp, dyldt, ietyp, imats, iposp )
!.....如果试应力在屈服面内，试应力即为真应力
        IF( dyldt .LE. 0.0D0 ) RETURN
        IF( DABS( dyldt - dyldp ) .LT. 1.0D-8 * drght ) RETURN
!.....否则进行应力缩减循环
        rfact = dyldt / ( dyldt - dyldp )
        reduc = 1.0D0 - rfact
        mstep = dyldt * 10.0D0 / drght + 0.5D0
        IF( mstep .GT. 20 ) WRITE( 12, 2000 )
        !F( mstep .GT. 50 ) nerrc = 594656
        IF( mstep .LE. 1 ) mstep = 2
        IF( nerrc .NE. 0 ) RETURN
        DO istrs = 1, 6
          strst( istrs ) = strsg( istrs ) + dstrs( istrs ) * reduc
          dstrs( istrs ) = dstrs( istrs ) * rfact / mstep
          dstrn( istrs ) = dstrn( istrs ) * rfact / mstep
        END DO
!.....开始应力缩减循环
        itype = 1
        CALL DMatM001( dmate, ietyp, imats, iposp, 1 )
        DO istep = 1, mstep
!         计算当前步的等效应力、流动矢量、塑性阵
          DO istrs = 1, 6
            strst( istrs ) = strst( istrs ) + dstrs( istrs )
          END DO
          hardk = HardM423()
          CALL PFlowVectorM423( pflow )
          CALL FlowVectorM423( strst, strsp, sflow, ietyp, imats, iposp )
          dlamd = Lamda( dmate, strst, strsp, dstrn, sflow, pflow, hardk,      &
                         ietyp, itype )
          DO istrs = 1, 6
            DO jstrs = 1, 6
              strst( istrs )          = strst( istrs ) - dlamd *               &
              dmate( istrs, jstrs ) * sflow( jstrs )
            END DO
          END DO
          CALL YieldM423( strst, strsp, yield, ietyp, imats, iposp )
          bcnst = drght / ( drght + yield )
          acnst =-yield / ( drght + yield )
          DO istrs = 1, 6
```

```
            strst( istrs ) = strst( istrs ) + acnst *                    &
                             ( strst( istrs ) + strsp( istrs ) )
          END DO
        END DO
!.....计算应力增量并恢复位移增量原值
        DO istrs = 1, 6
          dstrs( istrs ) = strst( istrs ) - strsg( istrs )
          dstrn( istrs ) = dstrb( istrs )
        END DO
2000    FORMAT( //2x, '警告：您选择载荷增量太大，计算结果可能有较大的误差。' )
        END

        SUBROUTINE YieldM423( strsg, strsp, yield, ietyp, imats, iposp )
!.....
!       模块功能
!           计算等效应力，屈服函数
!       strsg(6) — Gauss 点应力
!.....
        USE CtrlData
        USE MeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION strsg( 6 ), strsp( 6 ), strst( 6 )
        DOUBLE PRECISION kpara

        DO istrs = 1, 6
          strst( istrs ) = strsg( istrs ) + strsp( istrs )
        END DO
        ROOT3 = DSQRT( 3.0D0 )
!.....计算常量和应力不变量
        CALL StressState( strst, ietyp, smean, varj2, varj3, thita,      &
                          steff, qstrs )
        IF( nerrc .NE. 0 ) RETURN
!.....获得材料参数
        alpha = props( iposp + 2, imats )
        belta = props( iposp + 3, imats )
!.....计算屈服函数
        yleft = 3.0D0 * alpha * smean + steff
        yrigh = belta
        yield = yleft - yrigh
        RETURN
        END

        SUBROUTINE FlowVectorM423( strsg, strsp, flowv, ietyp, imats, iposp )
!.....
```

```
!       模块功能
!           计算流动矢量(屈服函数对应力矢量的偏导数)
!.....
        USE CtrlData
        USE ElmtData
        USE MeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION flowv( 6 ), vect1( 6 )
        DIMENSION strsg( 6 ), vect2( 6 )
        DIMENSION strsw( 6 ), vect3( 6 )
        DIMENSION strsp( 6 )
        INCLUDE 'Parameter.FD'
        DOUBLE PRECISION kpara

!.....初始化
        DO istre = 1, 6
          flowv( istre ) = 0.0D0
          vect1( istre ) = 0.0D0
          vect2( istre ) = 0.0D0
          vect3( istre ) = 0.0D0
          strsw( istre ) = strsg( istre ) + strsp( istre )
        END DO
!.....获得材料参数
        alpha = props( iposp + 2, imats )
        belta = props( iposp + 3, imats )
!.....计算应力张量不变量
        CALL StressState( strsw, ietyp, smean, varj2, varj3, thita,          &
                          steff, qstrs )
!.....根据应力状态确定流动矢量
        SELECT CASE( ietyp )
          CASE( MAT_SIMPLE_TENSIL )
!           简单拉伸应力状态
            vect1(1) = 1.0D0
          CASE( MAT_SIMPLE_SHEAR )
!           简单剪切应力状态
          CASE( MAT_PLANE_STRESS_AXISYMETRIC )
!           平面轴对称应力状态
          CASE( MAT_PLANE_STRESS )
!.....平面应力状态
          CASE( MAT_PLANE_STRAIN )
!.....平面应变（空间轴对称）应力状态
            strsw(1) = strsw(1) - smean
            strsw(2) = strsw(2) - smean
            strsw(4) = strsw(4) - smean
```

```
            vect1(1) = 1.0D0
            vect1(2) = 1.0D0
            vect1(3) = 0.0D0
            vect1(4) = 1.0D0
            vect2(1) = 0.5D0 * strsw(1) / steff
            vect2(2) = 0.5D0 * strsw(2) / steff
            vect2(3) =         strsw(3) / steff
            vect2(4) = 0.5D0 * strsw(4) / steff
            vect3(1) = strsw(2) * strsw(4) + varj2 / 3.0D0
            vect3(2) = strsw(1) * strsw(4) + varj2 / 3.0D0
            vect3(3) =-strsw(3) * strsw(4) * 2.0D0
            vect3(4) = strsw(1) * strsw(2) -                                    &
                       strsw(3) * strsw(3) + varj2 / 3.0D0
!.....
            cnst1 = alpha
            cnst2 = 1.0D0
            cnst3 = 0.0D0
          CASE( MAT_3D_PLATE_SHELL )
!           板壳应力状态
          CASE( MAT_3D_BLOCK )
!           一般复杂应力状态
          CASE DEFAULT
            WRITE( 12, 2000 )
            nerrc = 230101
        END SELECT
!.....计算流动矢量
        DO istre = 1, 6
          flowv( istre ) = cnst1 * vect1( istre ) +                             &
                           cnst2 * vect2( istre ) +                             &
                           cnst3 * vect3( istre )
        END DO
2000    FORMAT( //2x, '致命错误：无效的单元应力类型!' )
        END

        SUBROUTINE PFlowVectorM423( pflow )
!.....
!       模块功能
!           计算塑性流动矢量(屈服函数对塑性应力矢量的偏导数)
!.....
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION pflow( 6 )
        DO istrs = 1, 6
          pflow( istrs ) = 0.0D0
        END DO
```

```
      END

      FUNCTION HardM423()
!.....
!     模块功能
!         计算屈服函数对内时变量的偏导数)
!.....
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      HardM423 = 0.0D0
      END

      SUBROUTINE HistroyM423( strng, dstrn, strsg, dstrs, strsp, ietyp,          &
                              ielem, ielmx, imats, iposh, iposp )
      USE CtrlData
      USE MeshData
      USE GlobData
      USE ExtraMesh
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION strsg( 6 ), strng( 6 ), estrn( 6 )
      DIMENSION dstrs( 6 ), dstrn( 6 ), strsp( 6 ), dmate( 6, 6 )
      CALL StrsM423( strng, dstrn, strsg, dstrs, strsp, ietyp,                   &
                     ielem, ielmx, imats, iposh, iposp )
      CALL DMatM001( dmate, ietyp, imats, iposp, 1 )
      nstrg = NumOfStress( ietyp, 1 )
      CALL Invers( dmate, 6, nstrg, nerrc )
      DO istrg = 1, nstrg
        estrn( istrg ) = 0.0D0
        DO jstrg = 1, nstrg
          estrn( istrg )          = estrn( istrg ) +                             &
          dmate( istrg, jstrg ) * dstrs( jstrg )
        END DO
      END DO
      DO istrg = 1, nstrg
        ipose = iposh + istrg - 1
        IF( ielmx .LE. 0 ) THEN
          histr( ipose, ielem ) = histr( ipose, ielem ) +                        &
          dstrn( istrg          ) - estrn( istrg          )
        ELSE
          histx( ipose, ielem ) = histx( ipose, ielem ) +                        &
          dstrn( istrg          ) - estrn( istrg          )
        END IF
      END DO
      END

      SUBROUTINE DMatM423( dmatx, strsg, strsp, ietyp, imats, iposp )
```

```
USE CtrlData
USE MeshData
IMPLICIT DOUBLE PRECISION( a-h, o-z )
DIMENSION strsg( 6 ), dmatx( 6, 6 )
DIMENSION sflow( 6 ), dmate( 6, 6 )
DIMENSION pflow( 6 ), strsp( 6 )

itype = 1
cohes = props( iposp + 2, imats )
angle = props( iposp + 3, imats )
CALL YieldM423( strsg, strsp, yield, ietyp, imats, iposp )
IF( yield .LT.-cohes * DCOS( angle ) * 1.0D-5 ) THEN
  CALL DMatM001( dmatx, ietyp, imats, iposp, 1 )
ELSE
  hardk = HardM423()
  CALL PFlowVectorM423( pflow )
  CALL DMatM001( dmate, ietyp, imats, iposp, 1 )
  CALL FlowVectorM423( strsg, strsp, sflow, ietyp, imats, iposp )
  CALL DEPMatix( dmate, dmatx, strsg, strsp, sflow, pflow, hardk,        &
                 ietyp, itype )
END IF
END

SUBROUTINE KFunM423( attrn, thita, angle, kpara, dkdth )
IMPLICIT DOUBLE PRECISION( a-h, o-z )
DOUBLE PRECISION kpara

sinph = DSIN( angle )
sinth = DSIN( thita )
costh = DCOS( thita )
tanth = DTAN( thita )
sin3t = DSIN( thita * 3.0D0 )
cos3t = DCOS( thita * 3.0D0 )
tan3t = DTAN( thita * 3.0D0 )

ROOT3 = DSQRT( 3.0D0 )
IF( DABS( thita ) .LE. attrn ) THEN
  kpara = costh - sinth * sinph / ROOT3
  dkdth =-sinth - sinph * costh / ROOT3
ELSE
  sgnth = DSIGN( 1.0D0, thita )
  const = sgnth * costh * sinph / 3.0D0 / ROOT3
  apara = costh + sinth * tan3t / 3.0D0 +                                &
          const * tan3t - 3.0D0 * const * tanth
  bpara = sgnth * sinth + sinph * costh / ROOT3
```

```
      bpara = bpara / cos3t / 3.0D0
      kpara = apara - bpara * sin3t
      dkdth =-3.0D0 * bpara * cos3t
    END IF
    END
```

参考文献

[1] P.S.Symonds, T.X.Yu. Counterintuitive behavior in a problem of elastic-plastic beam dynamics. *J.Appl. Mech. ASME,* 1984,52:225-251

[2] S.H.Lee, S.S.Hsieh. Applocations of a self-adaptive algorithm to non-linear finite element analysis. *Int.J.Num.Methods Engrg*, 1989, 32:1057-1077

第10章

接触问题的有限单元法

工程中存在各种各样的接触问题，如齿轮啮合、汽机叶片榫槽连接，滚珠轴承、螺栓连接，铆钉连接，以及在金属板冲压成型过程中板模间接触等。这些接触问题可分成弹性接触问题、弹塑性接触问题、耦合热弹性接触问题、可变形固体和液体之间的接触问题等。Heinnrich Herty于1881年发表了关于接触问题的著名论文，奠定了接触力学的理论基础，影响深远。由于接触问题常常存在着复杂的工况和边界条件，很难采用经典的解析方法求解。

随着计算机技术的发展和有限元理论及应用的日趋完善，使得接触问题的实际求解成为可能，到上个世纪的七十年代，接触问题的数值方法进入实用阶段。求解接触问题的常用方法有：有限元法、边界元法、数学规划法以及形状优化方法等。其中，有限元法是分析接触问题最有效的方法之一。由于接触问题的复杂性，本章首先给出了接触问题的一些基本概念，然后介绍接触问题有限单元分析的一些常用方法。

10.1 接触问题有限单元法的基本概念

固体力学问题的基本方程包括平衡方程（运动方程），几何方程、本构方程（应力应变关系）以及相应的边界和初始条件（定解条件）。前面两章介绍了几何非线性和材料非线性问题的有限单元法，指出材料非线性是由本构方程的非线性性质引起的，几何非线性主要是由几何方程的非线性性质引起的。本章讨论的接触问题也是一种典型的非线性问题，其非线性是由接触边界上边界条件非线性引起的。

实际接触问题往往伴随材料非线性和（或）几何非线性，为简化问题，便于读者掌握接触问题的非线性本质和求解方法，本章主要讨论小变形弹性接触问题。

10.1.1 接触表面非线性

接触问题属于不定边界问题，即使是简单的弹性接触问题也具有非线性，其中既有由接触面积变化而产生的非线性以及由接触压力分布变化而产生的非线性，也有由摩擦作用产生的非线性。由于这种非线性和边界不定性，所以一般说来，接触问题的求解是一个反复迭代的过程。

当接触内力只和受力状态有关而和加载路径无关时，即使载荷和接触压力之间的关系是非线性的，仍然属于简单加载过程或可逆加载过程。通常无摩擦的接触问题就属于可逆加载问题。当接触面间存在摩擦时，在一定条件下可能出现不可逆加载过程或称复杂加载过程，这时一般要用载荷增量方法求解。

为简化分析过程，分析接触问题时一般采用如下基本假定：

（1）接触表面几何上是光滑连续的曲面；

（2）接触表面摩擦作用服从库仑定律；

（3）接触表面的力边界条件和位移边界条件均可用节点参量描述；

（4）接触表面的弹性流体动力润滑作用通过摩擦系数来体现。

工程实践表明，许多问题都可以在符合上述假定的前提下求得符合实际的解答。

10.1.2 接触面条件

对于接触问题，除了其场变量需要满足固体力学基本方程，以及相应的定解条件外，还必须满足接触面上的接触条件。接触条件主要包括两个方面，一是接触体之间在接触面上的变形协调性，不可相互侵入；二是摩擦条件。

对于接触或将要接触的两个物体，其界面接触状态可以分成分离、粘结接触和滑动接触三种。对于这三种情况，接触界面的位移和力的条件是各不相同的，而实际的接触状态又往往在此三种状态间相互转化，从而导致接触问题的高度非线性特点。

图 10.1 表示两个物体 I 和 II 相互接触的情形。通常称物体 I 为接触体（contactor），物体 II 为目标体或靶体（target），分别用 Γ_C^{I} 和 Γ_C^{II} 表示物体 I 和 II 的接触表面。称 Γ_C^{I} 为从接触面（被动接触面），称 Γ_C^{II} 为主接触面（主动接触面）。为便于给出三种状态下力和位移的边界条件，在接触面上任意点建立右手坐标系 $x_1'x_2'x_3'$，如图示。其中，坐标轴 x_1' 和 x_2' 位于接触点的切平面内，且相互正交，x_3' 沿接触点的法线方向并指向接触体 I。具体确定过程可参见退化板壳单元局部坐标系的建立。分别记坐标系 $x_1'x_2'x_3'$ 的三个单位基矢量为 $\boldsymbol{i}_1', \boldsymbol{i}_2', \boldsymbol{i}_3'$，则 $\boldsymbol{i}_3' = \boldsymbol{n}$。其中，$\boldsymbol{n}$ 为物体 II 在接触点的单位外法向矢量。沿局部坐标系的位移、应变和应力矢量分别记为

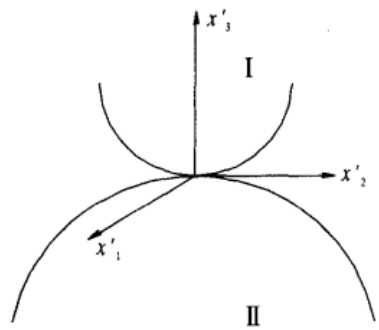


图10.1 接触物体及其坐标系

$$\boldsymbol{u}' = \{u_1', u_2', u_3'\}^T \tag{10.1}$$

$$\boldsymbol{\varepsilon}' = \{\varepsilon_{11}', \varepsilon_{22}', \varepsilon_{33}', 2\varepsilon_{23}', 2\varepsilon_{31}', 2\varepsilon_{12}'\}^T \tag{10.2}$$

$$\boldsymbol{\sigma}' = \{\sigma_{11}', \sigma_{22}', \sigma_{33}', \sigma_{23}', \sigma_{31}', \sigma_{12}'\}^T \tag{10.3}$$

如果考虑几何非线性，必须采用其他的应力、应变度量，对应局部坐标系下的矢量表达式可以类似给出。局部坐标系下的位移分量可以用单位基矢量表示为

$$u_k' = (\mathbf{i}_k')^T \boldsymbol{u} \qquad k = 1,2,3 \tag{10.4}$$

其中，$\boldsymbol{u} = \{u_1, u_2, u_3\}^T$ 为整体坐标系下的位移矢量。接触点对的三个局部坐标方向的接触面力和间隙分别为

$$p_k'^{\mathrm{I}} = -\sum_{j=1}^{3} \sigma_{kj}^{\mathrm{I}} n_j \qquad k = 1,2,3 \tag{10.5}$$

$$p_k' = p_k'^{\mathrm{II}} = \sum_{j=1}^{3} \sigma_{kj}^{\mathrm{II}} n_j \qquad k = 1,2,3 \tag{10.6}$$

$$g_k' = u_k'^{\mathrm{I}} - u_k'^{\mathrm{II}} + d_k' = (\boldsymbol{i}_k')^T (\boldsymbol{u}^{\mathrm{I}} - \boldsymbol{u}^{\mathrm{II}}) + d_k' \qquad k = 1,2,3 \tag{10.7}$$

式中，$u_k'^{\mathrm{I}}$ 和 $u_k'^{\mathrm{II}}$ 分别为局部坐标系中物体 I、II 上对应接触点（构成接触点对）的位移分量，$\boldsymbol{d}'=\{d_1', d_2', d_3'\}^T$ 为接触点对间的初始间距，其中，$d_1'=d_2'=0$。σ_{ij}^{I} 和 $\sigma_{ij}^{\mathrm{II}}$ 分别对物体 I、II 上对应接触点的应力分量。

下面列出接触点处三种状态对应的力和位移边界条件。

一、分离状态

位移条件

$$g_3'>0 \tag{10.8}$$

面力条件

$$\boldsymbol{p}'^{\mathrm{I}}=\boldsymbol{p}'^{\mathrm{II}}=\boldsymbol{0} \tag{10.9}$$

二、粘结接触状态

位移条件

$$g_3'=0 \tag{10.10}$$

$$g_1'=g_2'=0 \tag{10.11}$$

面力条件

$$\boldsymbol{p}'^{\mathrm{I}}+\boldsymbol{p}'^{\mathrm{II}}=\boldsymbol{0} \tag{10.12}$$

$$p_3'=-p_3'^{\mathrm{I}}=p_3'^{\mathrm{II}}<0 \tag{10.13}$$

$$\sqrt{(p_1')^2+(p_2')^2}<-\mu p_3' \tag{10.14}$$

三、滑动接触状态

位移条件

$$g_3'=0 \tag{10.15}$$

面力条件

$$\boldsymbol{p}'^{\mathrm{I}}+\boldsymbol{p}'^{\mathrm{II}}=\boldsymbol{0} \tag{10.16}$$

$$p_3'=-p_3'^{\mathrm{I}}=p_3'^{\mathrm{II}}<0 \tag{10.17}$$

$$\sqrt{(p_1')^2+(p_2')^2}=-\mu p_3' \tag{10.18}$$

上述各式中，μ 为摩擦系数。

采用有限元法分析接触问题时，需要分别对接触物体进行有限元网格剖分，并规定在初始接触面上，两个物体对应节点的坐标位置相同，形成接触点对。整体和局部坐标系下，两个物体由于接触荷载引起的等效节点力矢量分别记为

$$\bar{\boldsymbol{F}}^{\mathrm{I}}=\{\bar{F}_1^{\mathrm{I}}, \bar{F}_2^{\mathrm{I}}, \bar{F}_3^{\mathrm{I}}\}^T \qquad \bar{\boldsymbol{F}}'^{\mathrm{I}}=\{\bar{F}_1'^{\mathrm{I}}, \bar{F}_2'^{\mathrm{I}}, \bar{F}_3'^{\mathrm{I}}\}^T$$

$$\bar{\boldsymbol{F}}^{\mathrm{II}}=\{\bar{F}_1^{\mathrm{II}}, \bar{F}_2^{\mathrm{II}}, \bar{F}_3^{\mathrm{II}}\}^T \qquad \bar{\boldsymbol{F}}'^{\mathrm{II}}=\{\bar{F}_1'^{\mathrm{II}}, \bar{F}_2'^{\mathrm{II}}, \bar{F}_3'^{\mathrm{II}}\}^T$$

利用坐标变换关系可得

$$\bar{\boldsymbol{F}}'^{\mathrm{I}}=\boldsymbol{T}^T\bar{\boldsymbol{F}}^{\mathrm{I}} \qquad \bar{\boldsymbol{F}}'^{\mathrm{II}}=\boldsymbol{T}^T\bar{\boldsymbol{F}}^{\mathrm{II}} \tag{10.19}$$

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{i}_1' & \boldsymbol{i}_2' & \boldsymbol{i}_3'\end{bmatrix} \tag{10.20}$$

如果考虑摩擦力，采用载荷增量法分析，对应的离散化接触条件如下。

一、分离节点对

$$\Delta\bar{\boldsymbol{F}}'^{\mathrm{I}}=-\Delta\bar{\boldsymbol{F}}'^{\mathrm{II}}=\boldsymbol{0} \tag{10.21}$$

二、粘结节点对

$$\Delta\bar{\boldsymbol{F}}'^{\mathrm{I}} = -\Delta\bar{\boldsymbol{F}}'^{\mathrm{II}} \tag{10.22}$$

$$\Delta\boldsymbol{u}^{\mathrm{I}} = \Delta\boldsymbol{u}^{\mathrm{II}} \tag{10.23}$$

三、滑动节点对

$$\Delta g_3' = 0 \tag{10.24}$$

$$\Delta\bar{F}_3'^{\mathrm{I}} = -\Delta\bar{F}_3'^{\mathrm{II}} \tag{10.25}$$

$$\Delta\bar{F}_1'^{\mathrm{I}} = -\Delta\bar{F}_1'^{\mathrm{II}} = \pm\mu\Delta\bar{F}_3'^{\mathrm{I}}\cos\alpha \tag{10.26}$$

$$\Delta\bar{F}_2'^{\mathrm{I}} = -\Delta\bar{F}_2'^{\mathrm{II}} = \pm\mu\Delta\bar{F}_3'^{\mathrm{I}}\sin\alpha \tag{10.27}$$

式中，摩擦力增量的方向取决于滑动方向。α 为滑动方向与 x_1' 轴间的夹角，在增量求解过程中，由接触点对的相对位移来判断，也可以利用上一增量步的摩擦力方向来判定。

由式（10.21）～（10.27）可见，无论接触点对处于何种接触状态，每一接触点对都有 6 个接触定解条件，这些定解条件足以求解接触问题。

以上讨论的是空间接触问题，对于平面接触问题，可以类似地给出三种接触状态下的接触面条件。可以证明，任意一种接触状态下，每一个接触点对都有 4 个接触定解条件。感兴趣的读者可以参考[1]。

10.1.3 接触状态判定条件

接触问题求解时一般首先假定可能接触区内各接触点对的接触状态，根据相应的接触定解条件求解该问题。但是，一般说来，开始假设的接触状态并不一定符合实际，因此，如何确定三种状态的分界点是接触问题求解中最基本的问题。在有限元的能量泛函中，这一问题可以归结为确定积分界限问题。而这一问题只能通过迭代过程才能求解，每次迭代均要逐个检查接触点对的接触状态是否需要修改。

表 10.1 列出的接触判定条件是一种简便可行的方案，但并不是惟一的方案，在有限元分析时要注意，为防止出现不稳定的迭代过程，一般不直接从分离状态过渡到滑动状态，也不直接从顺向滑动修改为逆向滑动状态，反过来亦然。

表 10.1 接触状态判定条件

接触状态		判定条件
加载前	加载后	
分 离	分 离	$g_3' > 0$
	接 触	$g_3' \le 0$
粘 结	分 离	$\bar{F}_3'^{\mathrm{I}} < 0$
	粘 结	$\bar{F}_3'^{\mathrm{I}} \ge 0;\ \left\|\bar{F}_1'^{\mathrm{I}}\right\| \le \mu\bar{F}_3'^{\mathrm{I}}\cos\alpha;\ \left\|\bar{F}_2'^{\mathrm{I}}\right\| \le \mu\bar{F}_3'^{\mathrm{I}}\sin\alpha$
	滑 动	$\bar{F}_3'^{\mathrm{I}} \ge 0;\ \left\|\bar{F}_1'^{\mathrm{I}}\right\| > \mu\bar{F}_3'^{\mathrm{I}}\cos\alpha;\ \left\|\bar{F}_2'^{\mathrm{I}}\right\| > \mu\bar{F}_3'^{\mathrm{I}}\sin\alpha$
滑 动	分 离	$\bar{F}_3'^{\mathrm{I}} < 0$
	粘 结	$\bar{F}_3'^{\mathrm{I}} \ge 0;\ \Delta\bar{F}_1'^{\mathrm{I}}\Delta g_1 > 0;\ \Delta\bar{F}_2'^{\mathrm{I}}\Delta g_2 > 0$
	滑 动	$\bar{F}_3'^{\mathrm{I}} \ge 0;\ \Delta\bar{F}_1'^{\mathrm{I}}\Delta g_1 \le 0;\ \Delta\bar{F}_2'^{\mathrm{I}}\Delta g_2 \le 0$

接触点对状态判定条件是接触状态分析中最方便、最简单的。但它有明显的局限性，只能用以分析滑动量较小的情况。对于接触体间有较大相对滑动的情况，须用点–线、点–面或面–面接触条件。这些接触判定条件要比点对判定条件复杂得多，这里不再详述，很多文献都有这个方面内容的介绍。

10.1.4　加载的不可逆性

如前所述，接触压力分布和荷载之间的关系是非线性的。但在一定条件下，这种过程仍然是可逆的，可以认为属于简单加载状态。通常无摩擦的接触，其加载过程是可逆过程。对于有摩擦的接触问题，可能出现加载不可逆过程。

以如图 10.2 示结构为例，分析物体 II 上点 S 的接触力。记法向接触力为 p，是常数。切向接触力 q 不仅和外力 Q 有关，而且与是否存在 P 及 P 和 Q 的加载历史有关。考虑以下两种特殊情况。第一种情况是只加 P 而没有荷载 Q，则显然 $q=0$。第二种情况是只加 Q 而没有荷载 P，则 I、II 两物体在其接触面上没有约束作用，则 $p=0$ 且 $q=0$。

如果先加 P，再加荷载 Q，最后卸去水平方向荷载 Q，则其最终受荷情况和上述第一种特殊情况相同。但因加载历史不同，其接触状况却不一定相同，即 q 可能不等于零。图中给出了一种可能的切向接触力的变化过程。由图可见，虽然 $t=t_0$ 和 $t=t_5$ 时刻结构承受的外载相同，但接触状况却不相同，说明了加载的不可逆性。凡是可能导致接触内力不可逆性的加载过程均称为复杂加载过程。

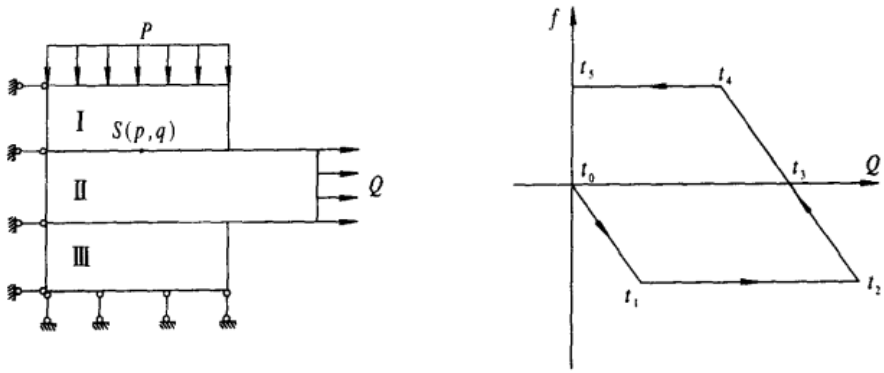


图 10.2　加载不可逆过程示意图

10.2 接触问题求解的一般过程

本节分析接触问题求解的一般过程，并简单介绍求解无摩擦弹性接触问题的一般有限元解法。

10.2.1　接触问题求解的一般过程

接触过程通常是依赖于时间，并伴随材料非线性和几何非线性的演化过程。特别是接触界面的区域和形状以及接触界面上运动学和动力学的状态也是事前未知的。这些特点决定了接

触问题通常采用增量法求解。第 4 章给出了大变形条件下的虚位移原理，第 8 章讨论了采用完全拉格朗日法（T.L.法）和更新拉格朗日法（U.L.法）增量求解大变形问题的方法，第 9 章给出了弹塑性有限元问题的求解方法。这些理论和方法同样适用于接触问题的求解。因此，如何将接触面条件适当地引入求解过程是接触问题求解的关键。

从上节的讨论可以看出，接触条件都是不等式约束，也称之为单边约束。另外，由于接触面的范围和接触状态也是事先未知的，接触问题只能采用通过迭代方法求解。每一增量步的迭代过程可以归纳如下。

(1) 根据前一增量步的结果和当前增量步给定的载荷条件，通过接触条件的检查和搜寻，假设此增量步第 1 迭代步求解时的接触面区域和接触状态。

(2) 根据上述接触区域和接触状态的假定，将接触面上的不等式约束改为等式约束作为定解条件引入方程并进行方程的求解。

(3) 利用接触面上的计算结果对假定的接触状态进行检查。如果接触面上每一点都不违反假定状态，则完成本增量步的求解并转入下一增量步的求解；否则修改接触状态，回到步骤(2)进行下一次迭代求解。

10.2.2 无摩擦弹性接触问题的一般有限元解法

针对无摩擦弹性接触问题，这里介绍一种最简单、直观的一般有限元解法，以便读者了解接触问题求解过程的特点及其与一般有限单元法间的联系。

当由两个弹性接触物体所组成的系统处于平衡状态时，根据虚功原理，分别写出两个物体的离散平衡方程，

$$\boldsymbol{K}^{\mathrm{I}}\boldsymbol{d}^{\mathrm{I}}=\boldsymbol{F}^{\mathrm{I}}+\bar{\boldsymbol{F}}^{\mathrm{I}} \tag{10.28}$$

$$\boldsymbol{K}^{\mathrm{II}}\boldsymbol{d}^{\mathrm{II}}=\boldsymbol{F}^{\mathrm{II}}+\bar{\boldsymbol{F}}^{\mathrm{II}} \tag{10.29}$$

式中，$\boldsymbol{K}^{\mathrm{I}}$和$\boldsymbol{K}^{\mathrm{II}}$分别为物体 I 和 II 的系统刚度矩阵；$\boldsymbol{d}^{\mathrm{I}}$和$\boldsymbol{d}^{\mathrm{II}}$为两离散体的节点位移矢量；$\boldsymbol{F}^{\mathrm{I}}$和$\boldsymbol{F}^{\mathrm{II}}$分别为作用在物体 I 和 II 上的外力等效节点力；$\bar{\boldsymbol{F}}^{\mathrm{I}}$和$\bar{\boldsymbol{F}}^{\mathrm{II}}$分别为两物体的接触力等效节点力。假定上式中已适当地引入了位移边界条件。

由于接触力等效节点力$\bar{\boldsymbol{F}}^{\mathrm{I}}$和$\bar{\boldsymbol{F}}^{\mathrm{II}}$是未知的，对于空间接触问题，每个接触点对增加了 6 个未知量；对于平面接触问题，每个接触点对增加了 4 个未知量。因此，在方程式(10.28)和(10.29)中，未知量个数多于方程数，无法直接求解，必须补充对应接触点对的接触定解条件。

对于无摩擦问题，首先假定各接触点对的接触状态，并代以相应的接触定解条件。引入接触定解条件后，可以由方程组(10.28)和(10.29)求出节点位移，进而可以求得接触点对上的接触力矢量。然后，根据接触点对的位移和接触力向量，依次按接触判定条件判断各点对的接触状态是否和假定的接触状态相符。如果不相符，则重新修改接触状态，重新将接触定解条件代入(10.28)和(10.29)式，再次迭代分析。经反复迭代计算，直到某次迭代计算前后接触状态完全一致为止。

以平面问题为例，假设物体 I 上的节点 i 和物体 II 上的节点 j 构成接触点对，系统的平衡方程为

$$\begin{bmatrix} \ddots & \vdots & \vdots & & \vdots & \vdots & \iddots \\ \cdots & K_{kk}^{\mathrm{I}} & K_{kl}^{\mathrm{I}} & \cdots & 0 & 0 & \cdots \\ \cdots & K_{lk}^{\mathrm{I}} & K_{ll}^{\mathrm{I}} & \cdots & 0 & 0 & \cdots \\ & \vdots & \vdots & & \vdots & \vdots & \\ \cdots & 0 & 0 & \cdots & K_{mm}^{\mathrm{II}} & K_{mn}^{\mathrm{II}} & \cdots \\ \cdots & 0 & 0 & \cdots & K_{nm}^{\mathrm{II}} & K_{nn}^{\mathrm{II}} & \cdots \\ \iddots & \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ u_i \\ v_i \\ \vdots \\ u_j \\ v_j \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \vdots \\ F_{xi} + \bar{F}_{xi} \\ F_{yi} + \bar{F}_{yi} \\ \vdots \\ F_{xj} + \bar{F}_{xj} \\ F_{yj} + \bar{F}_{yj} \\ \vdots \end{Bmatrix} \tag{10.30}$$

式中，$k=2i-1$，$l=2i$，$m=2j-1$，$n=2j$；F_{xi}、F_{yi} 和 F_{xj}、F_{yj} 分别为第 i 和 j 节点外荷载等效节点力；$\bar{F}_{xi}$、$\bar{F}_{yi}$ 和 $\bar{F}_{xj}$、$\bar{F}_{yj}$ 分别为第 i 和 j 节点接触力等效节点力。由于 $\bar{F}_{xi}$、$\bar{F}_{yi}$ 和 $\bar{F}_{xj}$、$\bar{F}_{yj}$ 是未知的，(10.30)式不能直接求解，需要补充接触面上的定解条件。

为便于引入这些定解条件，对(10.30)式进行坐标变换，将节点 i 和 j 的位移和等效节点力表示为局部坐标系中的分量形式，

$$\begin{bmatrix} \ddots & \vdots & \vdots & & \vdots & \vdots & \iddots \\ \cdots & K_{kk}'^{\mathrm{I}} & K_{kl}'^{\mathrm{I}} & \cdots & 0 & 0 & \cdots \\ \cdots & K_{lk}'^{\mathrm{I}} & K_{ll}'^{\mathrm{I}} & \cdots & 0 & 0 & \cdots \\ & \vdots & \vdots & & \vdots & \vdots & \\ \cdots & 0 & 0 & \cdots & K_{mm}'^{\mathrm{II}} & K_{mn}'^{\mathrm{II}} & \cdots \\ \cdots & 0 & 0 & \cdots & K_{nm}'^{\mathrm{II}} & K_{nn}'^{\mathrm{II}} & \cdots \\ \iddots & \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ u_i' \\ v_i' \\ \vdots \\ u_j' \\ v_j' \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \vdots \\ F_{xi}' + \bar{F}_{xi}' \\ F_{yi}' + \bar{F}_{yi}' \\ \vdots \\ F_{xj}' + \bar{F}_{xj}' \\ F_{yj}' + \bar{F}_{yj}' \\ \vdots \end{Bmatrix} \tag{10.31}$$

对于无摩擦接触问题，切向接触力等效节点力等于零，即 $\bar{F}_{xi}' = \bar{F}_{xj}' = 0$。如果节点 i 和节点 j 处于分离状态，则 $\bar{F}_{yi}' = \bar{F}_{yj}' = 0$，上式可以直接求解。如果节点 i 和节点 j 处于接触状态，则 $v_i' = v_j'$，$\bar{F}_{yi}' = -\bar{F}_{yj}'$。将式(10.31)的第 n 行加到第 l 行（包括右端项），再将系数矩阵的第 n 列加到第 l 列，并将第 n 行和第 n 列的所有元素赋值零，最后采用罚函数法引入约束 $v_i' = v_j'$，可得

$$\begin{bmatrix} \ddots & \vdots & \vdots & & \vdots & \vdots & \iddots \\ \cdots & K_{kk}'^{\mathrm{I}} & K_{kl}'^{\mathrm{I}} & \cdots & 0 & 0 & \cdots \\ \cdots & K_{lk}'^{\mathrm{I}} & K_{ll}'^{\mathrm{I}} + K_{nn}'^{\mathrm{II}} + \lambda & \cdots & K_{nm}'^{\mathrm{II}} & -\lambda & \cdots \\ & \vdots & \vdots & & \vdots & \vdots & \\ \cdots & 0 & K_{mn}'^{\mathrm{II}} & \cdots & K_{mm}'^{\mathrm{II}} & 0 & \cdots \\ \cdots & 0 & -\lambda & \cdots & 0 & \lambda & \cdots \\ \iddots & \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ u_i' \\ v_i' \\ \vdots \\ u_j' \\ v_j' \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \vdots \\ F_{xi}' \\ F_{yi}' + F_{yj}' \\ \vdots \\ F_{xj}' \\ 0 \\ \vdots \end{Bmatrix} \tag{10.32}$$

式中，λ 为罚系数，可取任意大于零的常数，为避免出现大数吃小数的现象，一般取与 $K_{ll}'^{\mathrm{I}}$ 和 $K_{nn}'^{\mathrm{II}}$ 相当的数量级。这样，式(10.32)就可以直接求解了。

这种有限元解法的特点是简单、直观。但存在严重的缺点：一是无法处理有摩擦的接触问题，二是只能一次加载，三是每次只要有任何一个点对不满足假定的接触状态，就必须重新修改整体刚度矩阵，增加运算工作量，降低了求解效率。这是一种早期提出的方法，现在已基本不用。

10.3 接触问题的罚函数法

前面已经提到，接触问题一般采用增量解法。假定 t_k 时刻两接触物体的状态已知，现求 $t_{k+1}=t_k+\Delta t_k$ 时刻接触问题的解。

10.3.1 接触问题的弱形式

将接触面约束条件引入势能泛函[2],可构造出如下形式的泛函，

$$\pi = U - W + G \tag{10.33}$$

式中，U 为应变能，W 为外力功，G 为接触面约束条件对应的约束项。于是问题的解相应于上述泛函的极值条件

$$\delta\pi = \delta U - \delta W + \delta G = 0 \tag{10.34}$$

其中，

$$\delta U = \int_V (\delta\varepsilon)^T \boldsymbol{\sigma}^{k+1} \mathrm{d}V \tag{10.35}$$

$$\delta W = \int_V (\delta\boldsymbol{u})^T \boldsymbol{f}^{k+1} \mathrm{d}V + \int_{\Gamma_t} (\delta\boldsymbol{u})^T \boldsymbol{t}^{k+1} \mathrm{d}\Gamma + \int_V (\delta\boldsymbol{u})^T \rho\ddot{\boldsymbol{u}} \mathrm{d}V \tag{10.36}$$

下面介绍将接触面约束条件引入势能泛函的拉格朗日乘子法和罚函数法。

一、拉格朗日乘子法

采用拉格朗日乘子法，则

$$G = \int_{\Gamma_c^{k+1}} (\boldsymbol{\lambda}'^{k+1})^T \boldsymbol{g}'^{k+1} \mathrm{d}\Gamma \tag{10.37}$$

式中，

$$\boldsymbol{\lambda}' = \{\lambda_1', \lambda_2', \lambda_3'\}^T \tag{10.38}$$

$$\boldsymbol{g}' = \{g_1', g_2', g_3'\}^T \tag{10.39}$$

称 $\boldsymbol{\lambda}'$ 为拉格朗日乘子。对(10.37)式求变分

$$\delta G = \int_{\Gamma_c} (\delta\boldsymbol{\lambda}')^T \boldsymbol{g}'^{k+1} \mathrm{d}\Gamma + \int_{\Gamma_c} (\boldsymbol{\lambda}'^{k+1})^T \delta\boldsymbol{g}' \mathrm{d}\Gamma \tag{10.40}$$

将上式分成两个部分，

$$(\delta G)_\lambda = \int_{\Gamma_c} (\delta\boldsymbol{\lambda}')^T \boldsymbol{g}'^{k+1} \mathrm{d}\Gamma \tag{10.41}$$

$$(\delta G)_u = \int_{\Gamma_C} (\delta\boldsymbol{g}')^T \boldsymbol{\lambda}'^{k+1} \mathrm{d}T \tag{10.42}$$

不难证明，$\boldsymbol{\lambda}'$ 为局部坐标系中接触面上的接触力矢量。

根据（10.7）式，可以导出

$$\delta g_i' = (\boldsymbol{i}_i')^T \delta(\boldsymbol{u}^{\mathrm{I}} - \boldsymbol{u}^{\mathrm{II}}) \tag{10.43}$$

考虑到 $\delta\boldsymbol{u}(\delta\boldsymbol{g}')$ 和 $\delta\boldsymbol{\lambda}'$ 的任意性，从（10.40）式可以得到

$$\delta U - \delta W + (\delta G)_u = 0 \tag{10.44}$$

$$(\delta G)_\lambda = 0 \tag{10.45}$$

对于粘结接触状态，λ_1'，λ_2' 和 λ_3' 都是未知的，由式（10.45）补充相应的方程数量。

对于无摩擦的接触情况，由于切向运动不受约束，此时 $\lambda_1'=\lambda_2'=0$。在有摩擦的滑动接触状态下，根据库仑定律可按下式确定 λ_1' 和 λ_2'，

$$\lambda_1'^{k+1}=-\mu\lambda_3'^{k+1}\left(u_1'^{\mathrm{I}\,k+1}-u_1'^{\mathrm{II}\,k+1}\right)\Big/u_T'^{k+1} \tag{10.46}$$

$$\lambda_2'^{k+1}=-\mu\lambda_3'^{k+1}\left(u_2'^{\mathrm{I}\,k+1}-u_2'^{\mathrm{II}\,k+1}\right)\Big/u_T'^{k+1} \tag{10.47}$$

其中，

$$u_T'^{k+1}=\sqrt{\left(u_1'^{\mathrm{I}\,k+1}-u_1'^{\mathrm{II}\,k+1}\right)^2+\left(u_2'^{\mathrm{I}\,k+1}-u_2'^{\mathrm{II}\,k+1}\right)^2} \tag{10.48}$$

从以上的讨论可见，对于滑动接触状态，不管是否有摩擦，都只有一个独立的拉格朗日乘子 λ_3'，式（10.45 ）补充了一个方程，即法向不可侵入约束。

用拉格朗日乘子法引入接触界面约束条件可以使约束条件得到精确的满足。不足之处是：（1）增加了方程的自由度数；（2）求解方程的系数矩阵中包含零对角元素，带来求解的不便。前一缺点带来的另一问题是使得包含惯性项的接触问题的有限元方程和显式时间积分的求解式不协调。

二、罚函数法

采用罚函数法将接触面约束条件引入泛函，此时

$$G=\frac{1}{2}\int_{\Gamma_c}(\boldsymbol{g}'^{k+1})^T\boldsymbol{\Lambda}'\boldsymbol{g}'^{k+1}\mathrm{d}\Gamma \tag{10.49}$$

式中

$$\boldsymbol{\Lambda}'=\mathrm{diag}\left(\alpha_1',\alpha_2',\alpha_3'\right) \tag{10.50}$$

其中 α_1'，α_2' 和 α_3' 为罚系数，为设定常数。于是

$$\delta G=\int_{\Gamma_c}\left(\delta\boldsymbol{g}'\right)^T\boldsymbol{\Lambda}'\boldsymbol{g}'^{k+1}\mathrm{d}\Gamma \tag{10.51}$$

容易证明

$$\overline{\boldsymbol{F}}'^{\mathrm{I}}=-\overline{\boldsymbol{F}}'^{\mathrm{II}}=-\boldsymbol{\Lambda}'\boldsymbol{g}' \tag{10.52}$$

对于无摩擦的接触状态，应当有

$$\alpha_1'=\alpha_2'=0 \tag{10.53}$$

对于有摩擦的滑动状态，根据库仑定律可得

$$\alpha_i'=-\mu\alpha_3'g_3'\left(u_i'^{\mathrm{I}}-u_i'^{\mathrm{II}}\right)\Big/u_T' \quad (i=1,2) \tag{10.54}$$

和拉格朗日乘子法相比较，用罚函数法引入接触界面约束条件的优缺点正好相反。它的优点是不增加问题的自由度，而且使求解方程的系数矩阵保持正定。因为不增加问题的自由度，可以和用显式数值积分方法求解包含惯性项的接触问题时的求解方程相协调。由于系数矩阵保持正定，在静力接触问题求解时，可以避免由于系数矩阵非正定性可能出现的麻烦。因此，罚函数法得到较广泛的应用。罚函数法的缺点是，约束条件只能近似地被满足。理论上，增大罚参数 $\boldsymbol{\Lambda}'$ 可使计算精度提高，但是接触问题中罚参数 $\boldsymbol{\Lambda}'$ 的大小往往受到更为严格的限制。一是罚参数 $\boldsymbol{\Lambda}'$ 愈大，显式解法的时间步长的临界值降低得愈多。二是罚参数 $\boldsymbol{\Lambda}'$ 过大，可能使相互接触的两个物体的相对运动发生虚假的反向，从而使解的过程不稳定。因此罚参数 $\boldsymbol{\Lambda}'$ 的具体取值是需要特别关注的问题。

三、其他方法

增广拉格朗日乘子法将上述两种方法合成，即取

$$G=\int_{\Gamma_c}(\boldsymbol{\alpha}'+\frac{1}{2}\boldsymbol{\Lambda}'\boldsymbol{g}')^T\boldsymbol{g}'\mathrm{d}\Gamma \tag{10.55}$$

相应的一阶变分为

$$\delta G=\int_{\Gamma_c}(\delta\boldsymbol{g}')^T(\boldsymbol{\alpha}'+\boldsymbol{\Lambda}'\boldsymbol{g}')\mathrm{d}\Gamma+\int_{\Gamma_c}(\delta\boldsymbol{\alpha}')^T\boldsymbol{g}'\mathrm{d}\Gamma \tag{10.56}$$

摄动拉格朗日法则在拉格朗日法的基础上加上一罚项，即

$$G=\int_{\Gamma_c}(\boldsymbol{\alpha}'^T\boldsymbol{g}'-\frac{1}{2\beta}\boldsymbol{\alpha}'^T\boldsymbol{\alpha}')\mathrm{d}\Gamma \tag{10.57}$$

其一阶变分为

$$\delta G=\int_{\Gamma_c}(\delta\boldsymbol{g}')^T\boldsymbol{\alpha}'\mathrm{d}\Gamma+\int_{\Gamma_c}(\delta\boldsymbol{\alpha}')^T(\boldsymbol{g}'-\frac{1}{\beta}\boldsymbol{\alpha}')\mathrm{d}\Gamma \tag{10.58}$$

10.3.2 罚函数法的有限元方程

以下以罚函数法为例说明接触问题有限元方程的建立和求解，与拉格朗日乘子法相关的内容可参见[3]。

一、接触块和接触点对

在运动过程中，两个物体的接触界面不仅区域大小是变化的，而且可能发生相互滑动。因此在对物体 I 和 II 进行有限元离散后，接触面两边 Γ_c^{I} 和 Γ_c^{II} 上的单元和节点的相互位置也是不断变化的。我们将单元处于接触面上的面（或边）称为接触块（或线），如图 10.3 示。对于一般的三维接触问题，每一个被（主）动接触块各与多个主（被）动接触块接触。图 10.4 表示一个被动接触块与四个主动接触块接触的情况。

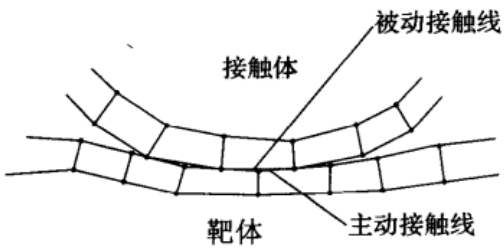


图 10.3 二维接触问题的主动、被动接触线

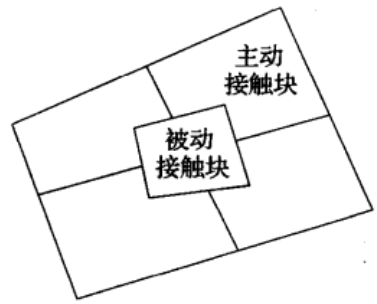


图 10.4 三维接触问题的主动、被动接触块

通常我们将被动接触块上的节点 P 和主动接触块上与其接触的 Q 点（注意不一定是节点）构成一个接触点对。它们在 t_{k+1} 时刻的坐标和位移分别是 x_P^{k+1}，u_P^{k+1} 和 x_Q^{k+1}，u_Q^{k+1}。因为 Q 点一般不是单元的节点，其坐标和位移可由所在接触块上节点的坐标和位移插值得到。现假设主动接触块是二维的 4 节点单元，则有

$$\boldsymbol{x}_Q=\sum_{i=1}^{4}N_i(\xi_Q,\eta_Q)\boldsymbol{x}_i \tag{10.59}$$

$$\boldsymbol{u}_Q = \sum_{i=1}^{4} N_i\left(\xi_Q, \eta_Q\right)\boldsymbol{u}_i \tag{10.60}$$

且有

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k \qquad \boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \Delta\boldsymbol{u}^k \tag{10.61}$$

（10.59）和（10.60）式中，N_i 是二维 4 节点单元的插值形函数，ξ_Q 和 η_Q 是 Q 点在接触块中的自然坐标，$\boldsymbol{x}_Q$ 和 $\boldsymbol{u}_Q$ 分别为整体坐标系中点 Q 的坐标和位移矢量，$\boldsymbol{x}_i$ 和 $\boldsymbol{u}_i$ 分别为整体坐标系中点 Q 所在接触块第 i 个节点的坐标和位移矢量。

这样一来，对于接触点对 P 和 Q 间的相对位移可以表示为，

$$\boldsymbol{u}_P - \boldsymbol{u}_Q = \boldsymbol{N}^c\boldsymbol{d}^c \tag{10.62}$$

其中

$$\boldsymbol{N}^c = \begin{bmatrix}\boldsymbol{I}_{3\times3} & -\boldsymbol{N}_1 & -\boldsymbol{N}_2 & -\boldsymbol{N}_3 & -\boldsymbol{N}_4\end{bmatrix}$$

$$\boldsymbol{d}^c = \begin{Bmatrix}\boldsymbol{u}_P^T & \boldsymbol{u}_1^T & \boldsymbol{u}_2^T & \boldsymbol{u}_3^T & \boldsymbol{u}_4^T\end{Bmatrix}^T$$

$$\boldsymbol{N}_i = N_i\boldsymbol{I}_{3\times3} \qquad (i = 1,2,3,4)$$

因为上式中 $\boldsymbol{d}^c$、$\boldsymbol{u}_P$ 和 $\boldsymbol{u}_Q$ 是在总体坐标系中定义的，为方便引入接触条件，需将其转换到局部坐标系中，即

$$\boldsymbol{u}'_P - \boldsymbol{u}'_Q = \boldsymbol{T}^T(\boldsymbol{u}_P - \boldsymbol{u}_Q) = \boldsymbol{T}^T\boldsymbol{N}^c\boldsymbol{d}^c \tag{10.63}$$

式中，$\boldsymbol{T}$ 是两种坐标系之间的转换矩阵，它的表达式见(10.20)式。

二、有限元方程

下面讨论三种接触状态下罚函数法的有限元方程。

（1）粘结接触状态

将(10.63)式代入(10.52)式，即可得到局部坐标系下一个接触点对的接触力引起的等效节点力向量，

$$\bar{\boldsymbol{F}}'^{k+1} = -\boldsymbol{\Lambda}'\boldsymbol{T}^T\boldsymbol{N}^c\boldsymbol{d}^c - \boldsymbol{\Lambda}'\boldsymbol{d}' \tag{10.64}$$

进一步可以得到整体坐标系下接触力等效节点力向量，

$$\bar{\boldsymbol{F}}^{k+1} = -(\boldsymbol{N}^c)^T\boldsymbol{T}\,\boldsymbol{\Lambda}'\boldsymbol{T}^T\boldsymbol{N}^c\boldsymbol{d}^c - (\boldsymbol{N}^c)^T\boldsymbol{T}\,\boldsymbol{\Lambda}'\boldsymbol{d}' \tag{10.65}$$

或者写成，

$$\bar{\boldsymbol{F}}^{k+1} = -\boldsymbol{K}_c\boldsymbol{d}^c + \tilde{\boldsymbol{F}}^{k+1} \tag{10.66}$$

其中，

$$\boldsymbol{K}_c = (\boldsymbol{N}^c)^T\boldsymbol{T}\,\boldsymbol{\Lambda}'\boldsymbol{T}^T\boldsymbol{N}^c \tag{10.67}$$

$$\tilde{\boldsymbol{F}}^{k+1} = -(\boldsymbol{N}^c)^T\boldsymbol{T}\,\boldsymbol{\Lambda}'\boldsymbol{d}' \tag{10.68}$$

由式(10.67)可以看出，$\boldsymbol{K}_c$ 为对称矩阵。

如果在计算中，取 $\alpha'_1 = \alpha'_2 = \alpha'_3 = \alpha$，则上式可以简化为

$$\boldsymbol{K}_c = \alpha(\boldsymbol{N}^c)^T\boldsymbol{N}^c \tag{10.69}$$

$$\tilde{\boldsymbol{F}}^{k+1} = -\alpha(\boldsymbol{N}^c)^T\boldsymbol{T}\,\boldsymbol{d}' \tag{10.70}$$

对所有的接触点对计算(10.66)式，并组集，则可得到系统的等效节点接触力向量。然后代入离散系统的平衡或运动方程，于是就得到如下求解方程，即

$$\boldsymbol{M}\ddot{\boldsymbol{d}}^{k+1}+\left(\boldsymbol{K}+\boldsymbol{K}_C\right)\boldsymbol{d}^{k+1}=\boldsymbol{F}^{k+1}+\bar{\boldsymbol{F}}^{k+1} \tag{10.71}$$

（2）摩擦滑动接触状态

根据（10.54）式和（10.63）式，在局部坐标中，可将摩擦滑动接触状态下对应的接触力表示为如下形式，即

$$\bar{\boldsymbol{F}}^{\prime 1}=-\boldsymbol{\alpha}_{fs}^{\prime}\left(u_3^{\prime 1}-u_3^{\prime \mathrm{II}}+d_3^{\prime}\right)=-\boldsymbol{\alpha}_{fs}^{\prime}\left(\boldsymbol{i}_3^{\prime T}\boldsymbol{N}^c\boldsymbol{d}^c+d_3^{\prime}\right) \tag{10.72}$$

其中

$$\boldsymbol{\alpha}_{fs}^{\prime}=\alpha_3^{\prime}\begin{bmatrix}-\mu\dfrac{u_1^{\prime}}{u_T^{\prime}} & -\mu\dfrac{u_2^{\prime}}{u_T^{\prime}} & 1\end{bmatrix}^T \tag{10.73}$$

利用(10.62)式，可得一个接触点对的等效节点力向量为

$$\bar{\boldsymbol{F}}^{k+1}=-\boldsymbol{N}_c^T\boldsymbol{T}\boldsymbol{\alpha}_{fs}^{\prime}\left(\boldsymbol{i}_3^{\prime T}\boldsymbol{N}_c\boldsymbol{d}_c+d_3^{\prime}\right) \tag{10.74}$$

注意到式（10.20），于是

$$\boldsymbol{T}\boldsymbol{\alpha}_{fs}^{\prime}=-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_1^{\prime}-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_2^{\prime}+\boldsymbol{i}_3^{\prime} \tag{10.75}$$

式（10.74）也可以写成式（10.66）的形式，此时

$$\boldsymbol{K}_c=(\boldsymbol{N}^c)^T\boldsymbol{T}\boldsymbol{\alpha}_{fs}^{\prime}\boldsymbol{i}_3^{\prime T}\boldsymbol{N}^c=\alpha_3^{\prime}\boldsymbol{N}_c^T(-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_1^{\prime}-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_2^{\prime}+\boldsymbol{i}_3^{\prime})\boldsymbol{i}_3^{\prime T}\boldsymbol{N}^c \tag{10.76}$$

$$\tilde{\boldsymbol{F}}^{k+1}=-(\boldsymbol{N}^c)^T(-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_1^{\prime}-\mu\frac{u_1^{\prime}}{u_T^{\prime}}\boldsymbol{i}_2^{\prime}+\boldsymbol{i}_3^{\prime})d_3^{\prime} \tag{10.77}$$

因此，系统的等效节点接触力向量以及有限元求解方程仍可用（10.71）式来表示，只是其中的$\boldsymbol{K}_c$和$\tilde{\boldsymbol{F}}$必须用（10.76）式和（10.77）式代入。还应注意到$\boldsymbol{K}_c$是非对称矩阵，这一点将为分析计算带来麻烦。

（3）无摩擦滑动接触状态

对于无摩擦滑动状态，$\mu=0$。令$\alpha_1^{\prime}=\alpha_2^{\prime}=0$，由（10.67）和（10.68）式得

$$\boldsymbol{K}_c=\alpha_3^{\prime}(\boldsymbol{N}^c)^T\boldsymbol{i}_3^{\prime}\boldsymbol{i}_3^{\prime T}\boldsymbol{N}^c \tag{10.78}$$

$$\tilde{\boldsymbol{F}}^{k+1}=-\alpha_3^{\prime}d_3^{\prime}(\boldsymbol{N}^c)^T\boldsymbol{i}_3^{\prime} \tag{10.79}$$

从（10.78）式可见，此时$\boldsymbol{K}_c$为对称矩阵。

同样，系统的等效节点接触力向量以及有限元求解方程仍可用（10.71）式来表示。

10.3.3 有限元方程的求解方法

瞬态接触问题有限元方程的求解方法常见的有两种，即显式解法和隐式解法。而常见的隐式解法是 Newmark 法，它的基本公式已在第 2 章介绍。下面讨论将 Newmark 法和 N-R 迭代法相结合求解接触问题有限元方程的方法。

将 Newmark 法应用于方程(10.71)，得

$$\hat{\boldsymbol{K}}\boldsymbol{d}^{k+1}=\hat{\boldsymbol{F}}^{k+1} \tag{10.80}$$

式中，

$$\hat{\boldsymbol{K}} = \boldsymbol{K} + \boldsymbol{K}_C + \frac{1}{\alpha \Delta t_k^2}\boldsymbol{M} \tag{10.81}$$

$$\hat{\boldsymbol{F}}^{k+1} = \boldsymbol{F}^{k+1} + \tilde{\boldsymbol{F}}^{k+1} + \boldsymbol{M}\left[\frac{1}{\alpha \Delta t_k}\dot{\boldsymbol{x}}^k + \left(\frac{1}{2\alpha} - 1\right)\ddot{\boldsymbol{x}}^k\right] \tag{10.82}$$

上式是一个非线性方程组，采用 N--R 迭代法计算，

$$\hat{\boldsymbol{K}}_T \Delta \boldsymbol{d}^{k(l)} = \hat{\boldsymbol{F}}^{k+1(l)} - \hat{\boldsymbol{K}}\boldsymbol{d}^{k+1(l)} = \boldsymbol{R}^{k+1(l)} \tag{10.83}$$

式中，

$$\hat{\boldsymbol{K}}_T = \boldsymbol{K}_T + \boldsymbol{K}_c^{k+1(l)} + \frac{1}{\alpha \Delta t_k^2}\boldsymbol{M} \tag{10.84}$$

$$\hat{\boldsymbol{F}}^{k+1(l)} = \boldsymbol{F}^{k+1} + \tilde{\boldsymbol{F}}^{k+1(l)} + \boldsymbol{M}\left[\frac{1}{\alpha \Delta t_k}\dot{\boldsymbol{x}}^k + \left(\frac{1}{2\alpha} - 1\right)\ddot{\boldsymbol{x}}^k\right] \tag{10.85}$$

式中，l 是迭代次数。这里的 $\boldsymbol{K}_c^{k+1(l)}$ 和 $\tilde{\boldsymbol{F}}^{k+1(l)}$ 也写成迭代形式，这是因为它们的表达式中，除罚参数 α' 和摩擦系数 μ 以外的各个量都依赖于 $\boldsymbol{d}^{k+1(l)}$。此迭代计算公式中应用了位移的增量迭代表达式，即

$$\boldsymbol{d}^{k+1(l+1)} = \boldsymbol{d}^{k+1(l)} + \Delta \boldsymbol{d}^{k(l)} \tag{10.86}$$

或者写成

$$\boldsymbol{d}^{k+1(l+1)} = \boldsymbol{d}^{k} + \sum_{j=0}^{l} \Delta \boldsymbol{d}^{k(j)} \tag{10.87}$$

并有

$$\boldsymbol{d}^{k+1(0)} = \boldsymbol{d}^{k} \tag{10.88}$$

以上各式中，$\boldsymbol{K}_c$ 对于不同接触状态的具体表达式可以分别参见（10.67）或（10.69）、（10.76）和（10.78）式。以下给出罚函数法递推迭代的计算步骤：

（1）将关心的时间段 $[0,T]$ 离散化，确定时间分点 t_k 和时间步长 Δt_k；

（2）计算总体质量阵 $\boldsymbol{M}$；

（3）给定初始位移 $\boldsymbol{d}^0$ 和初始速度 $\dot{\boldsymbol{d}}^0$，计算 $\boldsymbol{F}^0$ 和 $\tilde{\boldsymbol{F}}^0$；

（4）计算初始加速度 $\ddot{\boldsymbol{d}}^0$；

（5）令 $k=0$，$l=0$，并选择 Newmark 法中的参数 α 和 δ；

（6）计算 $\boldsymbol{F}^{k+1}$；

（7）形成整体切向刚度矩阵 $\boldsymbol{K}_T$；

（8）计算 $\hat{\boldsymbol{K}}\boldsymbol{d}^{k+1(l)}$；

（9）搜寻接触点对（P，Q）；

（10）判断所有接触点对的接触状态；

（11）计算 $\boldsymbol{K}_c^{k+1(l)}$，并形成 $\hat{\boldsymbol{K}}_T = \boldsymbol{K}_T + \boldsymbol{K}_c^{k+1(l)} + \dfrac{1}{\alpha \Delta t_k^2}\boldsymbol{M}$；

（12）计算 $\tilde{\boldsymbol{F}}^{k+1(l)}$，并形成 $\boldsymbol{R}^{k+1(l)}$；

（13）计算迭代步位移增量 $\Delta \boldsymbol{d}^{k(l)}$；

（14）更新 $\boldsymbol{d}^{k+1(l+1)}=\boldsymbol{d}^{k+1(l)}+\Delta\boldsymbol{d}^{k(l)}$，并迭代步的速度 $\dot{\boldsymbol{d}}^{k+1(l)}$ 和加速度 $\ddot{\boldsymbol{d}}^{k+1(l)}$；

（15）更新应力和接触力；

（16）校核接触条件，若发现不符合假定的接触点对，返回步骤（9），否则进行下一步骤；

（17）校核收敛准则，若未满足，则令 $l=l+1$，返回步骤（7）进行下一步迭代分析，否则进行下一步骤；

（18）令 $k=k+1$；

（19）若 $t_k<T$，则令 $l=0$，返回步骤（6）进行下一增量步分析，否则结束计算。

关于本节的求解方法，要注意以下两点：

（1）在递推迭代公式(10.83)中，若令 $\boldsymbol{M}=\boldsymbol{0}$，就可以用于不考虑惯性力影响的静态接触问题。

（2）隐式解法虽然是无条件稳定的，但时间步长 Δt_k 的取值仍应考虑计算精度和收敛性的要求。例如：

① 在用给定的 Δt_k 进行分析的迭代过程中，可能发现接触点对间的相互贯入超过某个允许值 ε_{d}（例如可取 ε_{d} 为最小单元边长的 5%和壳体厚度的 25%中的较小者），为了提高计算精度，在程序中可引入步长自动优化算法对这种现象进行处理。

② 当两个物体发生碰撞时，将激起高频的振动。如用隐式解法分析此类问题，为达到精度要求，仍应采用较小的 Δt_k。有时为了消除高频影响，还可以考虑引入数值阻尼（Newmark 方法中，取 $\delta>1/2$）。

10.4 接触问题的间隙单元法

应用有限单元法分析接触问题，一种最为直接、简单的方法就是在接触表面设立一种特殊形式的单元，通过这种单元与接触物体的连接、组装，建立接触系统整体平衡方程。上章介绍的节理单元就是这种类型单元之一。本节重点介绍分析接触问题的间隙单元法。

10.4.1 无摩擦问题的间隙单元法

间隙单元法首先用于处理无摩擦的接触问题，其基本思想是把接触边界之间的间隙区域处理成虚拟的单元。这样两个物体之间的接触系统（包含其间的间隙）就可以看成一个整体，都可以进行有限元网格剖分。我们把可能接触区中气体构成的单元成为间隙单元，这是一种具有特殊的材料非线性的单元，但在力的平衡关系和应变位移关系方面，它与物体 I 或 II 中的单元并无区别。未接触区域的间隙单元具有普通单元的形状，而已接触区的间隙单元在二维问题中为扁长四边形，其宽度趋于零，在三维问题中为一层薄膜，其厚度趋于零。每个间隙单元的节点一半在物体 I 上，另一半在物体 II 上。这种假想的由气体构成的单元，可以取 4～8 节点平面等参单元（通常有 4 或 6 个节点）或 8～20 节点空间等参单元（通常有 8 或 16 个节点），其刚度矩阵的形成方式和通常等参单元完全相同，但其弹性模量和剪切模量需要根据接触状态加以调整。这一点跟退化的板壳单元类似。因此，本节不再列出间隙有限单元列式，只着重讨论模量的调整过程。对于无摩擦接触问题，间隙单元的剪切模量 G 为零，因此只需讨论拉伸模量 E 的调整。

定义间隙应变

$$\varepsilon_M = \frac{u_3'^{\mathrm{I}} - u_3'^{\mathrm{II}}}{d_3'} \tag{10.89}$$

接触状态可用下述方法判断：

（1）当 $\varepsilon_M \geq 0$ 时，间隙展开，两物体继续分离。在这种情况下应调整间隙单元的刚度，使之不再阻止两物体相互分离的运动。也就是说应使弹性模量 E 很小，以至可以忽略不计。这时的间隙应力可表述为

$$\sigma_M = E\varepsilon_M \tag{10.90}$$

当间隙张开时，σ_M 也比物体 I、II 中的应力小得多。

（2）当 $-1 \leq \varepsilon_M < 0$ 时，间隙闭合，当 $\varepsilon_M = -1$ 时，间隙完全闭合，两物体已经接触，因此这时的间隙单元应有足够的刚度，以使其间隙应力 σ_M 和物体 I、II 上的应力具有相同的数量级。其 E 应和物体 I、II 的弹性模量相近。这一恰当的 E 值要通过多次迭代求得收敛解。

（3）当 $\varepsilon_M < -1$ 时，表明两物体已经相互嵌入，但嵌入是不允许的，因此必须调整模量使这些间隙单元的 ε_M 接近于–1。

间隙弹性模量的调整应遵循所谓应力不变原则，即要求在各次迭代中，间隙正应力尽可能保持不变。为此，首先要凭经验选定间隙正应力的初始值，由此可以确定初始弹性模量，其值应该既允许某些间隙单元完全闭合，又能限制物体的刚体位移。

间隙单元弹性模量的调整结果应保证：

（1）$\varepsilon_M > -1$ 的间隙单元的弹性模量应依次递减到某一很小的值。

（2）$\varepsilon_M = -1$ 的间隙单元，其弹性模量应调整到某一恰当的值，其数量和接触物体的弹性模量接近。

（3）没有任何单元的间隙应变 ε_M 明显的小于-1。

受压间隙单元弹性模量调整时所采用的应力不变原则可以表述为

$$E^{(k+1)}\varepsilon_g = E^{(k)}\varepsilon_M^{(k)} = 常数 \tag{10.91}$$

式中，ε_g 为两物体相互接触时的间隙应变。根据前面的分析，应取 $\varepsilon_g = -1$。于是，应力不变原则可具体表述如下，

$$E^{(k+1)} = -E^{(k)}\varepsilon_M^{(k)} \tag{10.92}$$

图 10.5 描述了按照以上迭代公式逐次逼近求得弹性模量近似解的过程。这里只是以一个单元为例说明其调整过程的，实际上各间隙单元的调整是同时进行的，其弹性模量和间隙应变是相互影响的。

只有同时满足如下条件时，才能认为接触迭代的求解过程已经收敛：

（1）所有受拉间隙单元的弹性模量均已趋于某一极小值。通常其值取为接触物体本身弹性模量的 $10^{-6} \sim 10^{-4}$ 倍。

（2）两次迭代之间的间隙应变已基本保持不变，即

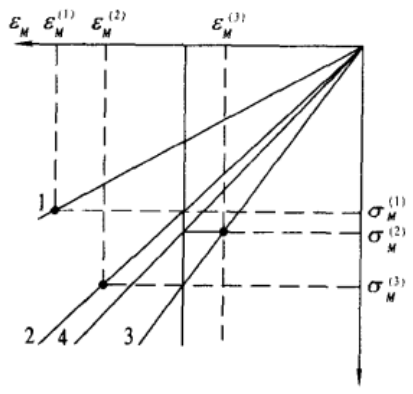


图10.5　弹性模量调整过程

$$\frac{\left|\varepsilon_M^{(k+1)}-\varepsilon_M^{(k)}\right|}{\left|\varepsilon_M^{(k)}\right|}\le 10^{-s} \tag{10.93}$$

其中，s 为由计算精度确定的负指数，可取 6～10。

（3）所有间隙单元均没有明显的相互嵌入现象，即

$$\left|\varepsilon_M^{(k+1)}+1\right|>10^{-t} \tag{10.94}$$

其中，t 也为由计算精度确定的负指数，通常取为 2。

10.4.2 有摩擦问题的间隙单元法

当接触物体表面之间存在摩擦时，必须同时调节间隙单元的拉伸模量 E 和剪切模量 G。间隙单元的刚度调节就是通过同时改变 G 和 E 实现的。在处于分离状态的区域，应使 $E^{(k)}=0$，$G^{(k)}=0$。对于两物体相互接触的区域，应使 $E^{(k)}$ 和物体本身的弹性模量相当，且两物体不会相互嵌入。处于粘结状态的间隙单元，两物体切线方向位移连续，剪切模量 $G^{(k)}$ 可根据 $E^{(k)}$ 和泊松比 ν 直接计算。处于滑动接触状态的间隙单元，两物体在切线方向发生滑动，则要根据库仑摩擦定律来调节剪切模量 G。

当接触表面之间存在摩擦时，其接触状态的判定要复杂一些。记间隙单元内的切向剪应力为 τ_M，其判定条件可以表述如下：

（1）当 $\varepsilon_M>-1$ 时，处于分离状态。

（2）当 $\varepsilon_M\le -1$ 且 $\tau_M<\mu\left|\sigma_M\right|$ 时，处于粘结接触状态。

（3）当 $\varepsilon_M\le -1$ 且 $\tau_M\ge\mu\left|\sigma_M\right|$ 时，处于滑动接触状态，滑动方向根据 τ_M 确定。

对于有摩擦的接触问题，其间隙单元弹性模量修正的过程也要复杂些。

（1）若两物体的某一可能接触区域处于分离状态，应使该区域内的间隙单元的拉伸模量和剪切模量按指数规律递减，但这一间隙单元拉伸模量的降低应当相当平缓，以保证迭代过程的收敛性。因为实际上各单元之间是相互联系、相互影响的，如果将处于分离状态的间隙单元的拉伸模量急速降低，非常容易导致不稳定的迭代过程。通常按以下公式取值，

$$E^{(k+1)}=\beta E^{(k)}\qquad G^{(k+1)}=\frac{1}{2}E^{(k+1)} \tag{10.95}$$

式中，β 为事先设定的常数，通常取 0.1～0.01。

（2）若两物体的某一可能接触区域处于粘结接触状态，则可按照前面所述应力不变原则修改拉伸弹性模量，即按（10.92）式迭代求得。然后根据 $E^{(k+1)}$ 来确定 $G^{(k+1)}$，例如可取

$$G^{(k+1)}=\frac{1}{2}E^{(k+1)} \tag{10.96}$$

（3）若两物体的某一可能接触区域处于滑动接触状态，其拉伸弹性模量的调整过程仍如前面所述，但剪切模量要根据库仑摩擦定律来调节。

当第 k 迭代步计算结果表明某接触点对处于滑动接触状态时，一般 $\tau_M^{(k)}>\mu\left|\sigma_M^{(K)}\right|$，而这不符合库仑摩擦定律，这种不符合实际的结果表明计算所得的剪应力 $\tau_M^{(k)}$ 太高。可以直观地认为，剪应力过大的原因是因为该处假设的剪切模量过大，所以修正时应减小剪切模量，以使

$\tau_M^{(k)} = \mu\left|\sigma_M^{(K)}\right|$。所以剪切模量的修正公式为

$$G^{(k+1)} = G^{(k)} \frac{\mu\left|\sigma_M^{(K)}\right|}{\tau_M^{(k)}} \tag{10.97}$$

10.4.3　间隙单元奇异性处理

当间隙法向正应变小于或等于–1 时，间隙单元的法线间隙便小于或等于零。这时如果直接按变形后的位置定义间隙单元，则这些单元都是奇异单元。形成刚度矩阵时，会发生雅可比行列式小于或等于零的情况。为避免奇异单元的出现，应在此处把物体相互拉开一些，以保证雅可比行列式大于零。拉开的距离一般很小，只要保证不发生除法浮点溢出即可。这种由嵌入后稍稍拉开产生的误差可以在迭代过程中消除。

几何修正的目的是防止接触边界相互嵌入，拉开的距离要保证不仅接触点对不相互嵌入，而且各单元的边界线也不相互嵌入。同时还要注意的是，这种距离拉开并不是边界线的平动，而是不均匀移动，嵌入量越大的地方移动量也越大。

这种几何修正完全是为了避免出现单元的奇异性，所以只在计算间隙单元的刚度矩阵时使用。

例10-1 受冲击作用的阻尼杆

图10.6示为一端固定的阻尼杆，杆的另一端受移动的刚性质量块的撞击。假定质量块是完全刚性的，其初始速度为 $v_0 = 100\,\text{mm/sec}$，质量 $M = 0.025\,\text{kg}$。阻尼杆长 $L = 100\,\text{mm}$，截面面积 $A = 1\,\text{mm}^2$，杨氏模量 $E = 10^7\,\text{N/mm}^2$，质量密度 $\rho = 0.00025\,\text{kg/mm}^3$。杆取瑞利阻尼，即

$$\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$$

本算例中，我们取 $\alpha = 0$，$\beta = 10^{-6}$。

杆被离散为四十个等长度的考虑几何非线性的杆单元，位移应变关系为

$$\varepsilon = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2$$

杆的自由端和刚性质量块之间用一间隙单元模拟两者间的碰撞接触，时间步长自动优化。

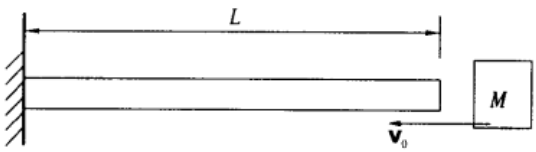


图10.6　受冲击作用的阻尼杆

图10.7和10.8给出阻尼杆自由端位移和应力的时间历程图，图10.9为优化时间步长随时间的变化曲线。作为比较，图10.10和10.11给出弹性杆（$\alpha = 0$，$\beta = 0$）自由端位移和应力的时间历程图，图10.12为优化时间步长随时间的变化曲线。

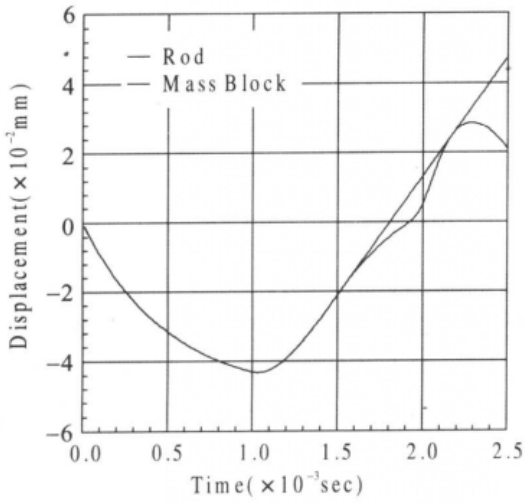


图10.7 阻尼杆自由端位移时间历程图

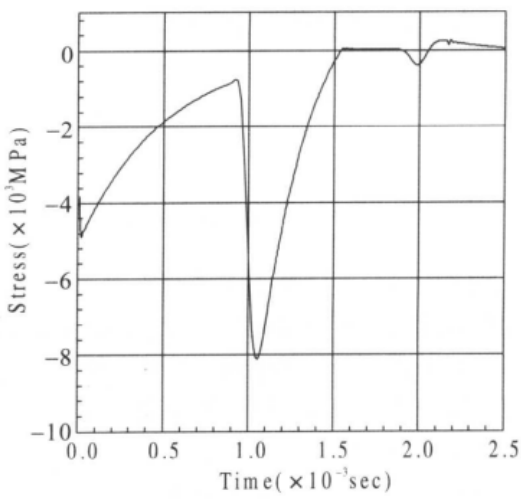


图10.8 阻尼杆自由端应力时间历程图

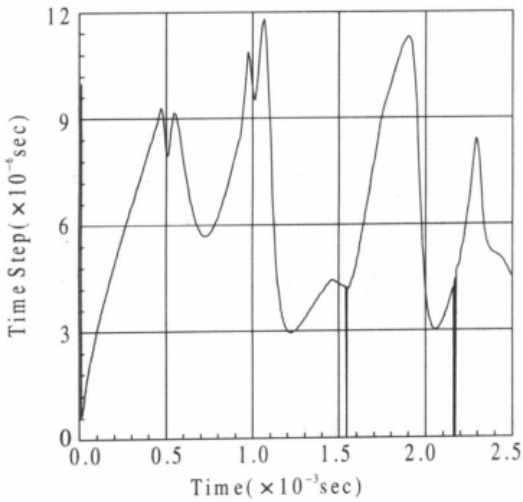


图10.9 阻尼杆自适应时间步长时间历程图

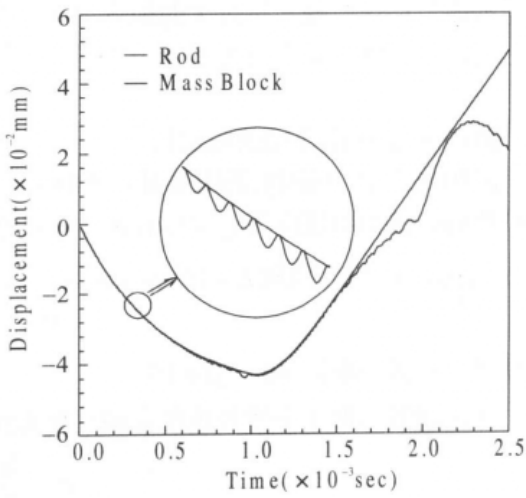


图10.10 弹性杆自由端位移时间历程图

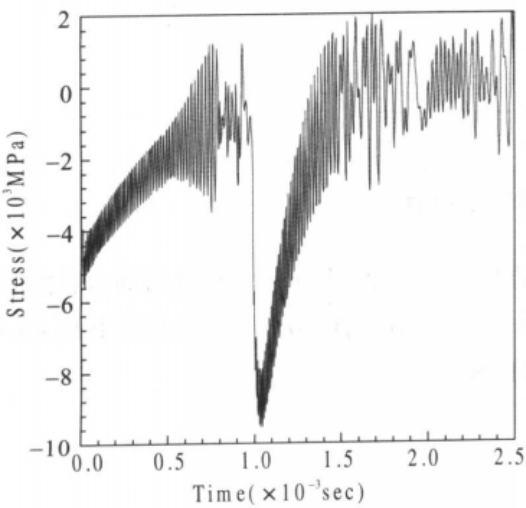


图10.11 弹性杆自由端应力时间历程图

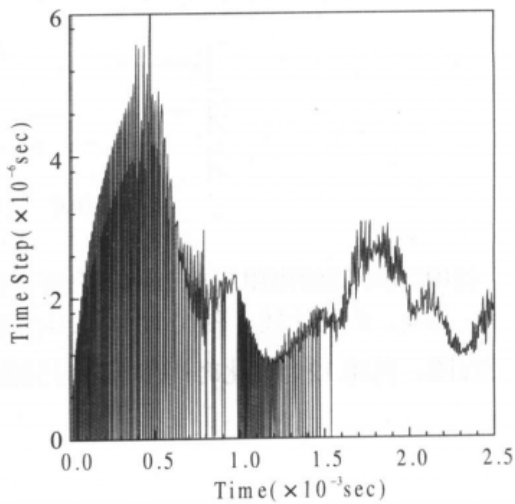


图10.12 弹性杆自适应时间步长时间历程图

由图不难看出，弹性杆件接触分析要比阻尼杆件接触分析复杂得多，自适应时间步长也随接触状态变化发生很大的改变，不断地出现很小的时间步长（图中几乎接近横轴的点）。因此，在动力接触问题分析中适当添加接触构件的阻尼是合适的。

参考文献

[1] 李润芳，龚剑霞编著.接触问题数值方法及其在机械设计中的应用. 重庆:重庆大学出版社，1991

[2] 丁皓江，何福保，谢贻权，徐兴.弹性和塑性力学中的有限单元法. 北京:机械工业出版社，1989

[3] 王勖成.有限单元法. 北京:清华大学出版社，2003

[4] 凌道盛.层合结构非线性有限元分析.浙江大学博士学位论文，1996

第11章

有限元软件方法

有限元软件是和有限元方法同时诞生，并且随着有限元方法和计算技术的发展而迅速发展的。根据不完全统计，半个世纪来已经研制了数百个通用的有限元软件和许多专用的有限元软件，它们在大中型科学和工程应用软件中占据相当大的比重。有限元软件是使用有限元方法解决土木、机械、造船、飞机、宇航、电机、冶金、核能、地震、物探、气象、渗流、水声、物理学、力学以及国防工程等各种科学和工程问题的关键，已经使有限元方法转化为直接推动科技进步和社会发展的生产力，发挥了巨大的经济和社会效益。

近 50 年来，有限元软件经历了有限元分析软件、有限元分析+CAD 软件、有限元分析+CAD+专家系统、智能性结构分析系统和集成化有限元软件环境等发展阶段。到目前为止，有限元软件的分析功能已经相当完备，其 CAD 功能已达到实用程度，但是它在人工智能和专家系统方面的功能仅仅是初步的，远未达到实用程度[1]。

从有限元软件的体系结构、实现方法和应用技术来看，现代有限元软件是一个多学科的、综合技术的集成化产品，除了有限元方法外，数据管理，用户接口，图形加工与管理，人工智能和专家系统，面向对象的软件设计，以及现代软件工程环境的方法和技术，在有限元软件研制中均占有重要的地位，它们和有限元方法相结合，已形成了一个特殊的称之为有限元软件技术的研究领域。在思维方法、表现形式和知识结构上，这个领域与有限元方法有明显区别，并且随着计算技术的发展它仍在迅速发展。

11.1 有限元软件发展概述

随着计算机软件和硬件技术的发展，有限元软件的发展大约可以分成如下六个阶段：

一、有限元分析软件；

二、有限元分析与设计软件；

三、有限元分析、设计与 CAD 软件；

四、有限元分析、设计与 CAD+专家系统；

五、智能性有限元结构分析系统；

六、集成化有限元软件开发环境。

受计算机硬件的限制，早期的有限元软件实质上只是有限元分析软件，其程序只由单元分析、组装和求解组成，缺乏前后处理功能。这一阶段的软件具有如下几大特点：

（1）整个软件是一个大程序，解题规模受机器内存限制；

（2）功能简单，能解决的问题范围窄，复杂性小；

（3）以批处理方法运行，缺少交互功能；

（4）输入方法繁琐，且仅限于简单数据；

（5）一般只能在大型机上运行，计算精度受硬件限制。

由于没有很好的前后处理功能，计算过程不能交互控制，软件使用者不仅要很好掌握有限元分析理论，还要对软件有深入的了解。通常，这类软件的用户即是软件的开发者。

20 世纪 70 年代中期以来，随着计算机运算速度的提高，内外存容量的扩大和图形设备的发展，以及软件技术的进步，有限元软件在原有的基础上扩充了与应用领域相关的前后处理功能。并引入文件管理技术，发展成为有限元分析与设计软件。这里的“设计”是根据有限元软件应用于结构工程设计时所增加的荷载组合、应力校核，按照规范的设计，以及绘制设计图等功能。虽然初期的前后处理的能力是比较弱的，但有限元软件朝着实用方向迈出了较为关键的一步。在前后处理软件中，计算机的交互技术得到发展和应用。

前后处理软件在有限元分析、设计与 CAD 软件阶段得到进一步发展和完善，并采用了数据库技术，以方便从 CAD 数据库中直接取出数据进行分析，并将分析结果传递给 CAD 系统。后处理软件的发展和 CAD 技术的应用，大大缩短了设计周期，提高了设计效率。在这样的软件系统支持下，设计人员可以进行多种方案的分析比较，以获得最优的设计方案。

有限元软件的发展，使得有限元软件的功能和应用领域进一步扩大，人们希望有限元软件能够具备有限元结构分析专家和有限元软件专家才能具有的解决实际问题的能力。AI 技术特别是专家系统技术的发展，使得这种要求得到了实现。有限元软件与专家系统的结合使得有限元软件发展成为有限元分析、设计与 CAD+专家系统。专家系统的引入，使得结构工程问题模型化、有限元软件的有效使用等原来只能靠专家经验解决的问题能通过软件引导用户自动实现，使得用户可以更方便、更容易、更正确地使用有限元软件解决科研和工程问题。

随着人工智能技术的发展，人们力图借助于该技术以完成对分析结果的评价，从而出现了所谓的智能性有限元软件。因为引入了人工智能方法，使得这类有限元软件可以更高速地完成建模和对结果的解释和评价工作，并且可以从结果的好坏入手，进行模型的改进和演化工作。所有这些都是通过软件自动完成的，从而大大降低了对用户的要求，解决问题的效率更高，也更能与 CAD 系统连接形成为一个整体。

有限元软件虽然具有很强的通用性，但是新材料、新方法层出不穷，应用领域不断扩大，与其应用领域中的其他功能软件的结合日趋紧密，从而对有限元软件的要求越来越高。特别是要求有限元软件具有较强的可裁剪性和再用性，可以很方便地与其他系统相连接，也可以容易地加入新的功能，这就要求有限元软件发展成为集成化的，可支持有限元方法应用和重新开发新的有限元软件的开发环境。国内著名学者崔俊芝在这方面做了一些开创性的研究工作。

在这种集成化有限元软件开发环境中，用户不但可以方便快速地完成特定问题的结构分析，而且可以裁剪出需要的成分，与某一领域特定的功能软件相结合形成具有新功能的其他软件系统，从而极大地推广了有限元结构分析软件在其他结构工程中的应用，使之充分发挥在结构设计中的工具作用。

有限元软件的发展推动许多行业的进步，国内外汽车工业和航空航天事业的发展历史见证了有限元软件的发展。

由于有限元软件前后处理及其相关知识超出了本课程内容，以下章节主要论述有限元分析软件的相关知识。

11.2 有限元分析的算法结构

11.2.1 算法结构和运算过程

从前面各章的讨论可以看出，一个通用有限元软件必然涉及各种各样的计算方法，如果按照有限元分析步骤划分，可以把它们分别归属 7 类，形成图 11.1 所示的算法结构。对于有限元分析软件而言，最关键的是单元库和算法库。通用的有限元分析软件的单元库应该包含各种各样的单元，而算法库则必须包含求解线性方程组、非线性方程组、初值问题、广义特征值等问题的算法。针对不同的有限元问题，使用这些算法可以组成不同的求解序列，图 11.2 为针对静态有限元问题的运算过程示意图。注意，即使是静态问题，非线性有限元算法也很多，如 Newton 迭代法、修正的 Newton 迭代法、增量解法等等。图中只是给出了一个基本算法的运算过程，而实际可组织的算法要丰富得多。

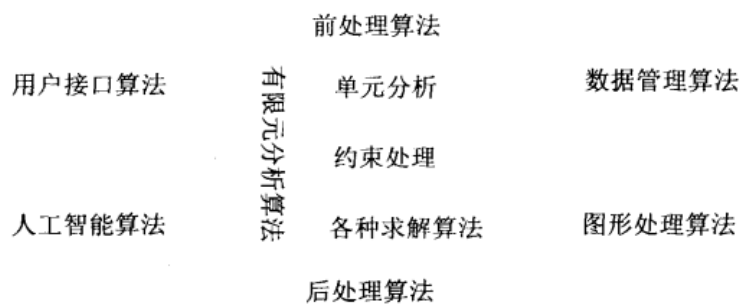


图 11.1 有限元软件的算法结构

11.2.2 单元分析

第 6 章中已经指出，单元形态是有限元分析的核心，它包括了有限元方法离散化的全部精华。它描述了构件的材料特征，单元的几何特征以及单元内逼近函数的形式和精度，因此，它是有限元分析成功与否的关键之一，是所有用有限元方法解决实际问题的人首先关注的。

经过近 50 年的发展，已经形成了十分丰富的单元形态库，可以说，无论是多么稀奇或复杂的问题，在现有单元形态库中，总可以找到适合分析它的单元形态。按基本构件形态，现行有效的单元形态可归纳为如下 11 类：

（1）杆梁单元：直杆、直梁和曲梁元；

（2）管单元：直管、弯管和 T 形管；

（3）平面元与板弯曲元：3 节点三角形、6 节点三角形、3～6 节点三角形、抗剪切面元（四边形）、4 节点四边形、8 节点四边形、4～8 节点四边形、18 节点等参数的高精度三角形、6 节点二次三角形和 6 节点五次三角形以及各类层合板单元；

（4）薄壳曲壳元：3 节点三角形、6 节点三角形和 4 节点四边形以及各类层合壳单元；

（5）轴对称单元：旋转壳曲线环、3 节点三角形环、6 节点三角形环、4 节点四边形环和 8 节点四边形环。

（6）断裂单元：三角形断裂单元、奇性断裂单元和裂隙单元等；

（7）接触单元：无摩擦接触单元、有摩擦接触单元和间隙接触单元等；

（8）厚壳元：简化到中面的厚壳元、6 节点三角形楔、12 节点三角形楔和 16 节点四边形棱柱，16～20 节点的六面体实体退化元等；

（9）空间实体元：4 节点四面体元、6 节点楔形体元、15 节点楔形体元、8 节点六面体元、20 节点六面体元和 8～20 节点六面体元；

（10）特殊单元：标量弹簧元、边界罚单元、6×6、12×12 广义单元和一般相关约束单元；

（11）过渡元：直杆与平面元过渡元、直杆与三维块体过渡元、梁与平面体过渡元和梁与三维体过渡元等。

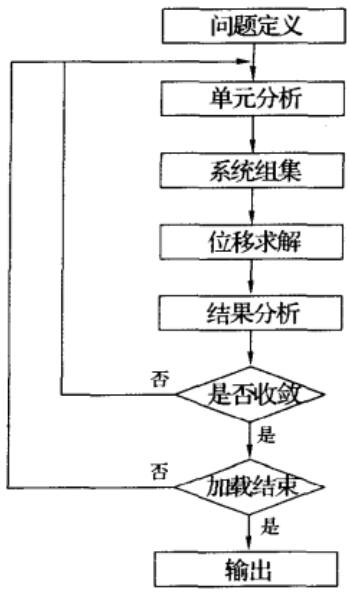


图 11.2 静态有限元问题的算法过程

有限元软件研制者在建立单元库时，必须严格地选择单元形态，应该选择那些经过多方考验，理论上和实践上都比较可靠的单元，切不可轻信某些单元形态提出者在其论文里所列出的优点，因为他的经验往往是很不充分的，例子是有倾向性的，而且往往是报喜不报忧的，难以根据他的意见对单元的优劣作出恰如其分的评价。在建立有限元软件的单元库时，一般应按如下原则选择单元形态：对结构类型的适应性；对构件几何形状的适应性；单元逼近精度；计算量；单元分析的复杂性。作为通用的有限元软件，要求单元库配备的单元形态能覆盖有限元软件应用领域的所有方面，即对于该软件所面临的问题集合中的每个具体问题，不管其几何形状与材料组合多么复杂，总可以在单元库里找到一些单元，使用它们可以对问题进行有效的离散化。因此，装配计算量小、精度高、适应性强又稳定可靠的单元库是开发有限元软件的一项十分重要的工作。为了保证有限元分析结果的精度，在选择单元形态时，除了按上面的原则综合考虑外，还应考察单元插值函数是否通过分片检验，是否具有坐标不变性，无伪零能模式，对网格畸变的灵敏度小和适应于不可压缩材料等。为了考核各类单元的适应性、可靠性和精度，1984 年 5 月在美国召开了第一次考核单元优劣标准的会议，自 1987 年起在英国出版了专门的评价有限元软件的杂志“BENCHMARKNAFEMS”，每期都以一定的篇幅评价单元形态。

11.2.3 有限元方程的组装与求解

在有限元软件的运行中，占用内外存空间最大的部分是组装单元分析的结果，形成总体有限元方程并存储之。单元分析结果和总刚度矩阵的存储格式不同，组装算法自然也会不同，无法一一详述。值得指出，有效地组装并存储大型有限元方程，合理地实施内外存调度，是有限元软件设计中的重要技术之一。

求解有限元方程是有限元分析的关键环节，也是计算量最大的计算步骤。实践表明，对于一般的有限元问题，求解有限元方程要花去 70% 左右的计算时间，因此求解方法的程序设计技巧的优劣直接影响着有限元软件的效率；同时求解方法也是影响有限元分析结果精度的重要环节，因为不同的求解方法，即使计算精度一样，其结果精度也是不一样的，因此应该针对有限元方程的存储格式精心地选择求解方法。对于有限元软件设计者而言，应该精心地选择高效能的解法。值得指出，随着分布式并行计算机系统的发展，发展高效能的有限元分析的并行算法已成趋势，目前主要是建立在区域分散策略基础上的并行多重子结构算法、分区多重网格和分区 PCGM 算法等。文献[1]列出了各种有限元方程、解法和存储方式。

11.3 国内外有限元软件简介

目前世界上流行的通用有限元软件有 300 多个，各种专用的有限元法程序则数以千计。但国内外著名的软件仅有几十个。下面简单介绍国内已经引进的国际上著名的有限元软件和国内的一些知名软件。

11.3.1 国外著名有限元软件介绍

目前国外著名的、国内已经引进的著名的商业软件有 SAP、ADINA/ADINT、ABAQUS、ANSYS、NASTRAN、MARC 等。

一、SAP

SAP(Structural Analysis Program)系统软件是一个线性静、动力分析的有限元程序。第一个版本于 1970 年 7 月发表，是由美国加州大学伯克利分校的 K.J.Bathe，E.L.Wilson，F.E.Peterson 等人共同研制的。SAP 软件经过了 SAP4、SAP5 等版本的发展，并实现了微机化。后来 SAP 主要由 Algor 公司进行维护和功能扩展，推出了基于 DOS 和 Windows 平台的 SuperSAP 系列软件。该系列软件增加了几何非线性、材料和接触非线性分析功能的模块，丰富了图形方式的前后处理模块。目前 SuperSAP 已被美国 50 个州以及世界近百个国家的有关研究机构和生产厂家所应用。

二、ADINA/ADINT

ADINA/ADINT 系统软件是非线性静、动/热应力分析程序。它是在 K.J.Bathe 教授的领导下，由麻省理工学院于 1975 年完成第一版。ADINA 是 NONSAP 软件的派生产品，现在由 K.J.Bathe 教授主导的 ADINA R&D，Inc 公司进行功能开发，具有强大的非线性分析功能，可实现流场、结构、热的耦合分析。

三、ABAQUS

David Hibbitt 博士是 ABAQUS 系统软件的奠基人。ABAQUS 系统软件具有几乎所有线性

和非线性分析功能，如静力、动力、热力耦合、刚体动力学、力电耦合以及使用隐式时间积分的非线性动态响应分析。能够进行设计灵敏度分析(DSA)，可以考虑波浪荷载、拖动和浮力以模拟海洋石油平台管道和电缆系统。它还能够分析结构的疲劳寿命和疲劳强度储备因子，确定部件的疲劳寿命。

四、ANASYS

ANASYS 是 20 世纪 60 年代末期，John Swanson 博士为了核能应用在 Westinghouse 开发的一个大型线性和非线性有限元程序。ANASYS 具有多场耦合分析的功能(压电、交变磁场、热、流体、声)，且界面友好。经过几十年的改进，ANASYS 已成为有着广泛用户的大型、通用有限元软件。

五、NASTRAN

NASTRAN 是 1966 年美国国家航空航天局(NASA)为满足航空航天工业对结构分析的需要，主持开发的大型有限元软件，1970 年公开了程序的第一个版本。后有多家公司对 NASTRAN 进行维护改进，推出了多个不尽相同的版本，如 MSC/NASTRAN、CSA/NASTRAN、UAI/NASTRAN、ME/NASTRAN、SAS/NASTRAN、COSMIC/NASTRAN 等。其中以 MSC 公司对原始的 NASTRAN 改进最多，添加了新的单元库，增强了程序的分析功能，改进了用户界面，提高运算精度和效率。其产品 MSC.NASTRAN 是目前美国联邦航空管理局(FAA)的工业标准的源代码程序，其分析功能覆盖了绝大多数工程应用领域，并为用户提供了方便的模块化功能选项。

11.3.2　国内有限元软件介绍

我国有限元软件研制起步于上世纪的 70 年代，第一个自行开发的、有自主版权的有限元软件是由大连理工大学钟万勰院士组织开发的 JIGFEX。由于加入了我国建筑结构设计规范，在土木工程领域内解决了大量的工程结构设计分析问题，中国建筑科学研究院研制开发的 PKPM 系列结构分析软件成为我国用户最多的有限元分析软件。在消化吸收美国 SAP 软件的基础上，北京大学袁明武教授领导和组织开发了 SAP84，曲圣年教授提出了 SAP5P。在 Taylor 主持开发的 FEAP(Finite Element Analysis Program)基础上，浙江大学徐兴教授推出了 ZD-FEAP。在 20 世纪 90 年代，吉林大学胡平教授领导开发了模具分析有限元软件系统 KMAS。这些软件都为我国有限元软件的开发和应用起到很大的促进作用，为我国培养了大量的有限元分析人员。

与国际上著名的有限元软件相比，我国的有限元软件系统还有很大的差距，我国的软件工作者还需要不断的努力。从 PKPM 成功的经验可以看出，结合一个具体的工程背景，以有限元为分析手段，开发专用的分析软件也是有限元软件发展的趋势之一。作者采用实体退化板壳单元开发了桩筏基础分析的专用软件 POGAP，获得较好的社会效益。

11.4　有限元分析软件中若干技术问题

有限元分析软件的编制是一个十分复杂的系统过程，涉及多学科的知识，存在大量的编程技巧，许多技巧需要在编程过程中去总结和体会。本节总结了编者对有限元软件开发过程中的一些技术问题的理解，供读者参考。

一、 软件开发系统环境

软件开发系统环境的选择是关系软件生命周期的一个关键因素，基于什么样的操作系统和采用什么样的编程语言是非常重要的。众所周知，DOS 已经退出微型计算机操作系统的行列，Windows2000、Windows XP 和 WindowsNT 是目前和将来一段时间居主导地位的操作系统，因此，原先在 DOS 操作系统下开发的软件，由于落后的编程思想、落后的内存管理和过时的文本界面而不合时宜，如不改进，势必被淘汰。而这样的改进需要大量的人力和物力，有时几乎是不可能的。如果这样，这些软件的生命就走到了尽头。

编程语言的选择则是软件开发过程中又一个重要课题。目前可选用的编程语言种类繁多，各有优点，常见的有 FORTRAN、BASIC、C、PASCAL 等等。一般认为严谨的 FORTRAN 语言适合于科学计算，具有较高的计算精度；C 语言则具有灵活方便，比较适合于界面和图形处理。BASIC 和 PASCAL 等语言在有限元分析软件中使用得不多。FORTRAN 语言又包含 MS-FORTRAN，NDP-FORTRAN 和 Visual FORTRAN 等开发环境，不同的开发环境遵循不同的语言标准。无论调试手段还是语言功能（内部函数）和语法结构，Visual FORTRAN 都具有无与伦比的优势。采用 Visual FORTRAN 作为有限元分析软件的开发环境是合适的。当然利用面向对象的 C＋＋开发有限元分析软件也是比较理想的选择之一，类、重构、继承等概念的使用使源代码更为精简，可读性更强。在现有条件下，C＋＋和 FORTRAN 混合编程也是一个容易实现的、合理的选择。动态链接库的广泛使用也让有限元分析软件的开发人员具有更多的选择。

二、变量命名规则

变量的命名往往被认为是一件微不足道的事情而不被注意。其实变量的命名十分重要，一个良好的变量命名规则使得程序更加便于阅读和维护。因此作者认为任何一个软件，特别是大型软件，在研制之前首先必须对软件中的变量命名规则进行约定，所有参与软件设计的编程人员必须严格按照规则对变量进行命名。

三、数组的管理

有限元程序设计中数组的内存管理是关系到程序稳定与否的关键因素，也是程序维护中最让人头痛的问题。目前中小型有限元程序（也包括各种自行研制开发的软件）常采用如下两类数组管理方式。

第一、固定大小的数组管理，即所有必要数据，如单元定义、节点坐标、材料信息等，均采用固定大小的数组存储。如

```
DIMENSION lnods(20,500)
```

这种数据管理有明显的缺点。当数组存储元素申请太多时，造成不必要的内存浪费，甚至可能降低计算效率；当数组存储元素申请太少时，则又限制了问题求解的规模，当需要存储的元素超过开设的数组界限时，如果软件没有设置完善的自检功能，可能会出现异常错误或得到错误的结果。由于有限元分析程序的规模一般很大，数组界限的任何修改都是一个细致、繁杂的工作，不仅工作量大，而且容易出错。

第二、相对的动态管理。这一数组管理方法首先为同一类型（如双精度型、浮点型、整型、逻辑型等）的所有数组开设一个足够大的内存空间，然后根据不同物理量需要的空间大小在相应的内存数组中依次占用空间。比如节点坐标可以在双精度型或浮点型数组中占用空间，单元定义则在整型内存数组中分配空间。这种管理方法显然要比上一方法进步得多。然而这一方法也有其缺点，第一，对于不同内存资源的计算机，为充分利用内存资源需要调整内存数组

的大小，并重新编译，连接；第二，程序设计相对比较复杂，不易阅读和维护；第三，很难确定各种类型变量需要空间的相对关系，内存资源仍有很大的浪费。

随着计算机软硬件技术的发展，在 FORTRAN 语言中使用可调数组可以克服前两种方法的不足。

第三、完全的动态管理。完全动态管理方法充分利用于 FORTRAN 90 中新增加的语言功能：动态可调数组（Allocatable Array）。可调数组的使用不仅充分利用了内存资源，而且使得程序设计更加合理、简洁，程序的可阅读性更强。下面以单元定义数组 lnods 在 Viusal FORTRAN 环境中的使用为例加以说明动态数组的使用步骤。

(一) 可调数据声明

```
ALLOCATABLE lnods(:,:)                !说明可调及维数。
```

(二) 必要时分配空间

```
ALLOCATE( LNODS( MNODE, NELEM ), STAT = nerrc )
IF( nerre .GT. 0 ) STOP 'No enough memory'
```

(三) 可调数组查询

```
IF( ALLOCATED( lnods ) ) … …
```

(四) 使用完毕后释放相应内存

```
DEALLOCATE（ lnods ）
```

这样，任何一个变量都可以有自己的说明，并在需要时为其分配内存资源，在不需要时又可以及时释放内存，以提高内存的利用效率。

四、数据传递

在调用和被调用的程序模块之间，往往需要传递大量信息。绝大部分 FORTRAN 程序采用有名或无名公用区（common 语句）和虚实结合的方式传递数据。根据编者多年的编程经验，虚实结合的方式不便软件维护，而且当传递参数太多时，可能会降低程序运算的效率，甚至造成堆栈溢出。公用区传递数据的方式虽然简洁，但也不易维护，对可调数组也不适用。在 FORTRAN 90 中，使用 MODULE 可以很好地解决这一问题。

MODULE 是 FORTRAN90 新增加的类似 C＋＋中类的程序模块，它可以定义类似成员变量和成员函数的变量、函数和子过程。与变量定义相似，无需实例化，这一点与类不同。这样，软件开发人员可以在 MODULE 中定义需要的变量，也可以利用 MODULE 内的函数在使用变量前将它们初始化，包括为可调数组开设空间、数组和变量的初始化，在分析结束或不再使用时及时释放可调数组的内存空间。如果需要在某个过程（SUBROUTINE）或函数（FUNCTION）中使用 MODULE 中的变量，只需在该过程或函数中使用 USE 语句即可。在编者研制的 VFEAP 程序中，所有数组均为可调数组，并分别归纳入不同的 MODULE 管理，可调数组内存的分配和释放均由 MODULE 内部的函数完成。所有的 MODULE 均保存在同一个文件中。可调数组和 MODULE 语句的综合使用不仅保证了内存资源的充分利用，而且使得数据的传递和维护更加简单、方便。下面列出的是 VFEAP 程序中的一个典型 MODULE，读者可以从中理解 MODULE 的使用方法。

```
MODULE MeshData
IMPLICIT DOUBLE PRECISION( a-h, o-z )
ALLOCATABLE coord(:,:), lnods(:,:)          ! 可调数组声明
ALLOCATABLE lslnt(:,:), vslnt(:,:)
```

```
    ALLOCATABLE lfixs(:,:), vfixs(:,:), llnks(:)
    ALLOCATABLE props(:,:), lmats(:)
    SAVE coord, lnods, lslnt, vslnt
    SAVE lfixs, vfixs, llnks, props, lmats

    CONTAINS
      SUBROUTINE InitMesh                          ! MODULE 内函数
        USE CtrlData
        IF( nerrc .GT. 0 ) RETURN
        ALLOCATE( coord( ndime, npoin ),
     +            lnods( mnode, nelem ),
     +            lslnt(       2, mslnt ), vslnt( mdofn, mslnt ),
     +            lfixs( mdofn, npoin ), vfixs( mdofn, npoin ),
     +            props( nprop, nmats ), lmats( nelem ),
     +            llnks( npoin          ), STAT = ierro )
        IF( ierro .GT. 0 ) nerrc = 3
        IF( nerrc .GT. 0 ) RETURN
        CALL InitFloat( coord, ndime * npoin, 9.9D9 )
        CALL InitFloat( vslnt, mdofn * mslnt, 0.0D0 )
        CALL InitFloat( vfixs, mdofn * npoin, 0.0D0 )
        CALL InitFloat( props, nprop * nmats, 0.0D0 )
        CALL InitInteger( lnods, mnode * nelem, 0 )
        CALL InitInteger( lslnt, 2       * mslnt, 0 )
        CALL InitInteger( lfixs, mdofn * npoin, 0 )
        CALL InitInteger( lmats, nelem, 0 )
        CALL InitInteger( llnks, npoin, 0 )
      END SUBROUTINE InitMesh

      SUBROUTINE DelMesh
        DEALLOCATE( coord, lnods, lslnt, vslnt, lfixs,
     +              vfixs, props, lmats, llnks )
      END SUBROUTINE DelMesh
    END MODULE MeshData
```

五、运算过程控制

本章一开始就指出，与有限元分析相关的算法有很多，不同的算法具有不同运算过程。许多复杂的工程问题往往需要采用不同的算法进行分析比较，以选择一种合适的方法。针对每一种算法开发不同的有限元程序显然是很不明智的。如何把众多的算法统一在同一个程序中，用户可以根据需要选择其中的一种，或者直接控制运算过程是一个相对比较复杂而又必须面对的

问题。

几乎所有成熟的商业化有限元软件都有自己的算法语言，然而良好的前处理往往将整个算法过程包装到只需点击简单选项、输入几个必要的参数就可以了。这对用户来说是一件好事，提高了计算分析的效率。但要想使用好商业软件，深入理解有限元分析程序的计算过程是有必要的。从编程的角度来看，利用算法语言控制算法过程的软件对模块化和数据管理的要求更加高，编程难度更大。

六、有限元分析程序的考核

有限元分析程序的研制开发是一个十分复杂的工作，难度很大。随着程序功能的不断完善，程序规模越来越大。程序的每一个环节都关系到软件分析的结果，因此必须确保每个模块的正确性。对软件的考核是必不可少的。

所谓考核就是通过对一些有“精确”解的、一般是简单的例子的分析和比较，评价被考核模块的有效性和正确性。这些“精确”解可以是解析解，也可以是其他已被公认正确的分析软件的计算结果。编者认为，有限元分析软件的考核可分成两大类，一是算法考核，二是单元考核。“新”算法在程序中实现后的考核必须是非常严格的、全面的，评价的标准不能是工程化的。两个分析软件采用完全相同的方法分析同一个算例，即使出现 1%的相对偏差也可能预示着其中某个软件存在问题。对新开发单元的考核必须十分严谨。考核人员必须全面了解单元的性态，合理地选择算例。比如，简单拉伸和纯剪切变形是考核 4-8 节点平面等参单元必不可少的算例，但如果用它去分析一根细长梁的弯曲变形，而厚度方向只有一个单元，则可能得不到考核人员期望的结果。读者不妨用 3 节点的三角形等参单元去分析图 11.3 示结构在自重作用下变形，仔细分析、对比两种不同网格给出的计算结果，一定会有所收获。

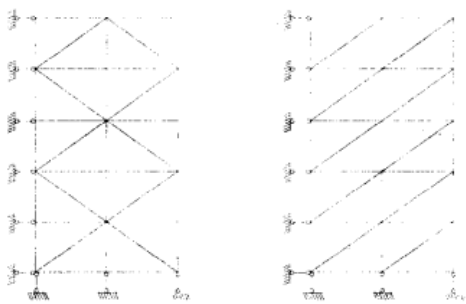


图 11.3　一个简单模型示意图

参考文献

[1] 崔俊芝，梁俊.现代有限元软件方法. 北京:国防工业出版社，1995

[2] 徐兴，郭乙木，沈永兴.非线性有限元及程序设计. 杭州:浙江大学出版社，1993

第12章

非线性有限元分析程序 VFEAP

本章重点介绍一个非线性有限元分析程序VFEAP。VFEAP是编者借鉴了国内外的一些有限元分析软件包（特别是**FEAP**和**ZD-FEAP**）的优点，自1991年开始研制的科研和教学程序。VFEAP以中文Windows98（或更高版本）和Virtual Fortran （6.5版以上）为开发平台，采用FORTRAN 90标准编制。

该程序最初的研制工作是围绕复杂航空结构的有限元分析展开的。此后，结合土木工程，该软件得到了进一步的发展，开发了一些特色单元，主要应用于桥梁工程和岩土工程问题的有限元分析。

VFEAP利用算法宏命令控制分析过程，能有效组织多种算法分析线性和非线性问题。我们希望该程序能为读者提供一个非线性有限元分析程序的基本框架，为进一步的研究开发提供一个平台。

12.1 VFEAP 的主要功能和结构

本节介绍VFEAP的主要功能、特点和程序结构（包括主要的子程序）。

一、主要功能和特点

VFEAP借鉴了FEAP和ZD-FEAP开发成功的经验，由编者自主开发的有限元分析程序，可以求解线性和非线性结构的下列问题：

（1）结构静力分析；

（2）广义特征值问题（固有频率和模态分析、临界荷载分析）；

（3）与时间相关的问题分析（蠕变问题、瞬态响应问题、固结问题等）。

分析程序采用波前法求解线性方程组，采用宏命令组织算法，目前可组织的算法有：

（1）Newton-Raphson法；

（2）修正的Newton-Raphson法；

（3）载荷增量法；

（4）Wilson-θ 法；

（5）Newmark法；

（6）矢量逆迭代法；

（7）Crank法。

VFEAP具有如下特点：

（1）完全基于Windows操作系统和Visual Fortran(6.5版以上)开发，采用模块化程序设计，可读性强，并具有良好的可移植性。

（2）通过Windows高效管理内存，利用可调数组充分使用内存。通过Windows操作系统，不仅可以方便地使用所有的计算机物理内存，而且使得虚拟内存的使用和管理更为简洁、方便。Fortran 90中可调数组的应用可以很好地避免内存资源的浪费，从而提高有限元分析的计算效率。

（3）采用改进的波前法求解线性方程组。该方法根据计算机的内存容量优化内外交换的时机和频率，从而兼顾了波前法的求解效率和求解规模。

（4）变量命名的规范化使得程序更加容易掌握，便于维护。

（5）算法宏命令最大限度地减少了程序开发人员间的相互干扰，也便于使用者更有效地组织各种有限元算法。

虽然VFEAP能够解决一些实际工程问题，但缺陷的存在总是不可避免，其功能和应用范围也需要不断地扩大。因此，需要众多的使用者在使用过程中不断地发现缺点、改进缺点，使之不断完善。

二、程序框图和结构

图12.1为VFEAP的程序框图。



图12.1　VFEAP程序框图

VFEAP为模块式结构，总体上分成两大块：Windows界面模块和有限元分析模块。其中Windows界面模块是所有Windows运行程序不可或缺的，它利用Virtual Fortran的资源功能设计界面，并完成图形界面的输入和输出操作。而有限元分析模块VFEAP负责完成有限元分析过程。

按调用关系，VFEAP的模块式结构可分成四层。第一层只有一个模块，即Windows程序的入口模块WinMain。第二层主要包含一个负责完成有限元分析的子程序VFEAP和若干实现图形界面输入输出的功能模块。第三层和第四层包含众多的子程序，这里不再一一说明。

有限元分析模块VFEAP调用的主要子程序如图12.2所示。这些子程序完成有限元模型的数据输入和检查，有限元分析过程控制以及计算结果的整理、分析和输出。由图可以看出，Execute

是VFEAP中最重要的模块。

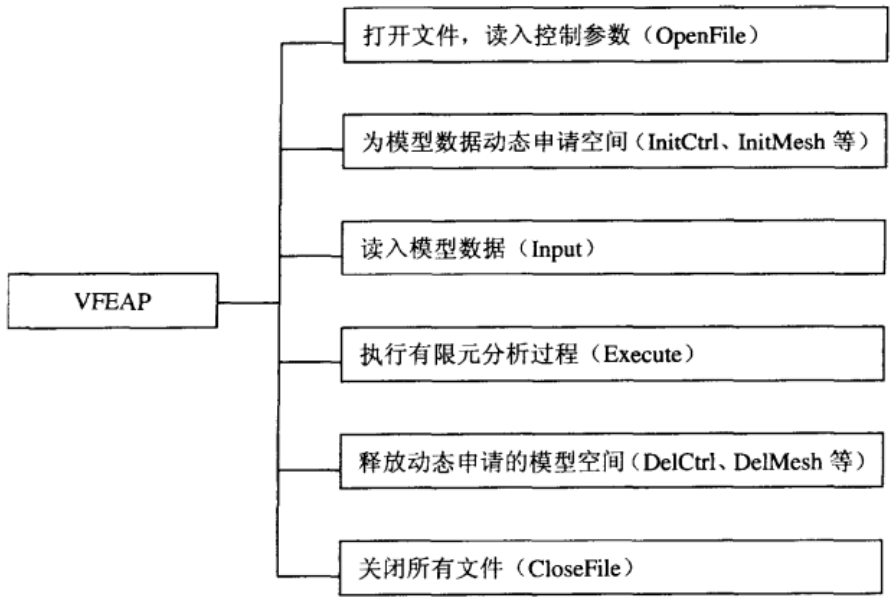


图12.2 VFEAP模块调用关系

Execute模块的调用关系如图12.3所示。波带宽优化、单元分析、总刚组集以及方程组求解等过程均由Excute在执行特定宏命令时，调用相应的模块完成。譬如，当Execute模块执行宏命令stiffen（计算总体刚度阵）时，它调用程序中的一个重要模块ElmLib来完成单元分析、单刚组集等相应功能。

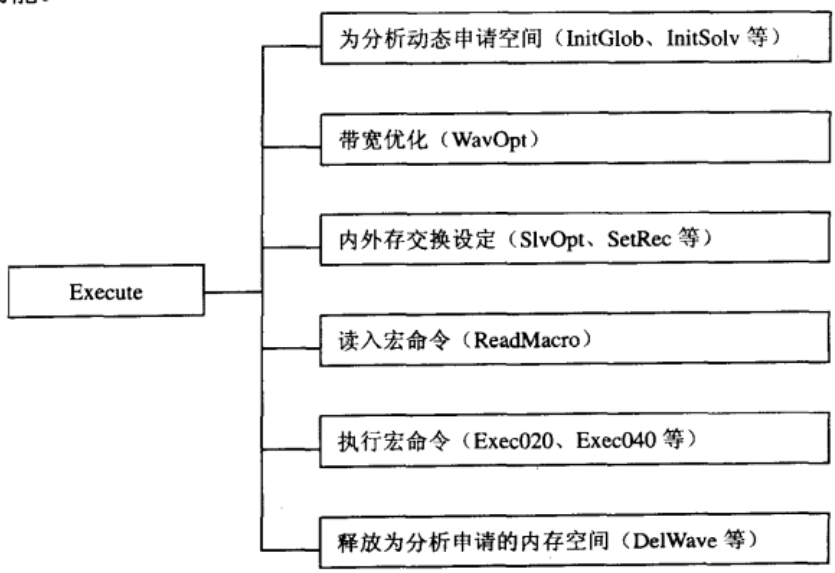


图12.3 Execute模块调用关系

ElmLib模块是一个单元库，其调用关系如图12.4示。子程序ElmLib的主要功能是：根据波阵带宽优化后的单元顺序，依次从有限元模型数据中提取当前单元在当前时刻对应的基本信息，如单元节点编号、单元节点坐标、单元节点位移、单元历史量等，并形成波阵信息，然后调用相应的单元进行单元分析，获得单元的元素矩阵或矢量，组集得到总体矩阵或矢量，并在必要时组织内外存交换。

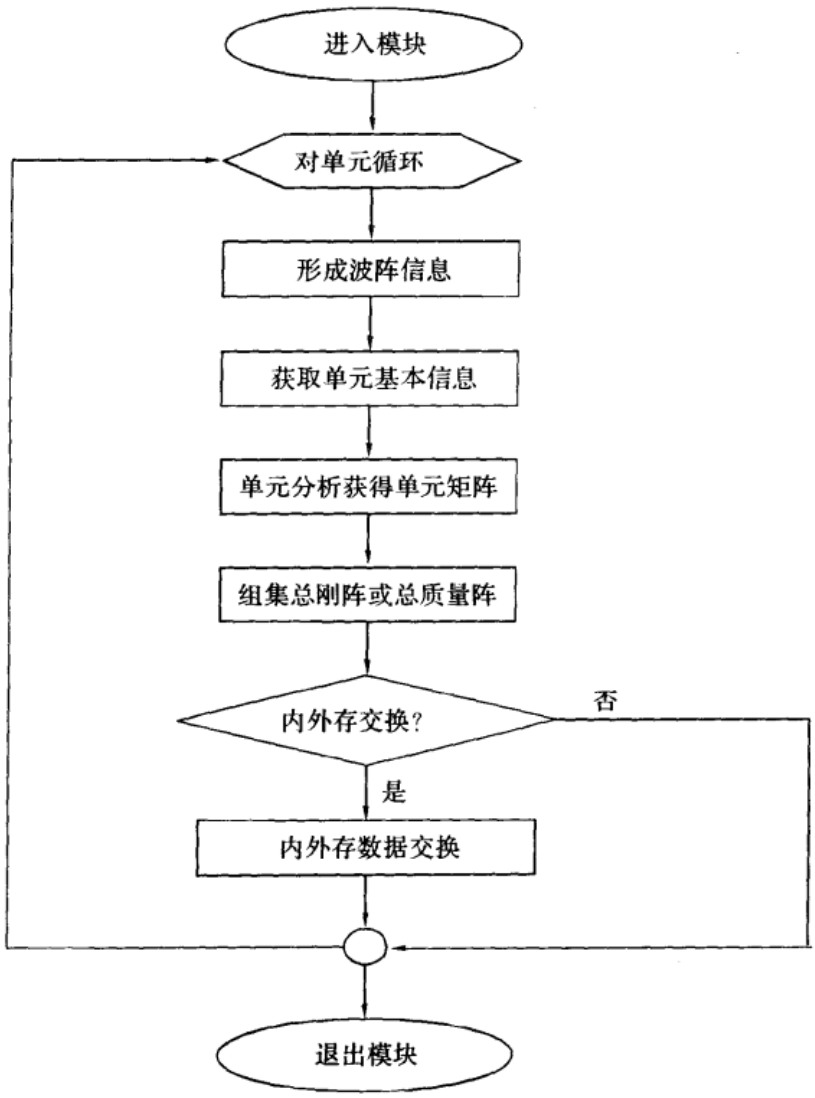


图12.4　ElmLib程序框图

12.2　VFEAP 的主要源文件和变量

12.2.1　主要源文件及其功能

整个VFEAP程序拥有80多个Fortran源程序、两个资源文件和若干工程文件。这里不可能列出所有的源文件，并对它们进行说明，下面仅列出其中最主要的部分源文件。

VFEAP.f90　　主程序所在源文件，包括WinMain、VFEAP、OpenFile、CloseFile等子程序。

Input.f90　　包含所用有限元模型数据输入所需要的子程序，包括节点信息、单元

	信息、材料信息、载荷信息、约束信息和初始条件等的输入。
Print.f90	包含所有数据输出所需要的子程序，包括节点位移、节点应力以及与后处理衔接的模型数据、节点位移、节点应力和单元属性等的输出。
Macro.f90	输入并解析宏命令，并为执行这些宏命令做好必要的准备工作。
Data.f90	数据模块文件。程序中所有全局的整型、浮点型变量和数组都在此文件中定义。它是一个非常重要的文件，是继续开发的基础，读者必须认真阅读此文件。
Dynamic.f90	与时间相关的求解方法涉及的子程序大多在此文件中，譬如Newmark法、Wilson-θ法、Crank法等。
EigenP.f90	利用矢量逆迭代法计算广义特征值问题涉及的所有子程序在此文件中。
Elmlib.f90	单元库文件，负责波阵信息和单元信息的准备、单元调用、组集等功能的实施。读者在添加新单元时会涉及此文件。
Execute.f90	负责计算过程中动态数组的申请和释放，依次执行读入的宏命令以完成用户定义的有限元分析过程和输入输出操作。
Front.f90	波前法求解线性方程组涉及的所有子程序都在这个文件中，包括波阵优化，内外存交换控制、分解与回代等。
Gauss.f90	在此文件中可获得各种数值积分方法的求积节点和求积系数。其中包括Gauss积分、Labatto积分、Hammer积分等。
Load.f90	计算节点力、节线（单元的边）分布力和节面（单元表面）分布力的等效节点力。
MainStrs.f90	分析一点的应力状态，计算该点的主应力、主应力方向、应力偏量、应力偏量不变量等。
Shape.f90	负责计算形函数及其关于自然坐标、整体坐标的导数。其中包括4～8节点四边形平面单元，8～20节点六面体实体单元，3～6节点三边形平面单元，6～15节点三棱柱单元，4节点四面体单元等的形函数。
Support.f90	处理位移约束条件。
Elmt×××.f90	单元子程序，其中×××代表单元序号。

12.2.2 变量的命名规则

为便于阅读和维护，程序在编写过程中遵循如下变量命名规则：

（1）所有变量名取五个字符，如 nmats，lmats 等。

（2）遵守 FORTRAN 语言的隐含约定，即以 I-N 字符开头的变量为整型变量，其余为浮点型变量。由于程序采用双精度计算，一般子程序都有以下隐含类型说明语句，

IMPLICIT DOUBLE PRECISION(a-h, o-z)

（3）整型变量的第一个字符代表变量属性。规定 m 代表最大值或最小值，n 代表总数，i，j 代表循环变量，k 代表累计变量。如 mnode 代表单元节点数的最大值，nnode 则代表当前单元的节点数目。

（4）整型变量的最后一个字母代表变量的指定范围。规定 s 代表整个有限元系统，e 代

表单元，n 代表节点。例如 ndofs 代表有限元系统的总自由度数，ndofe 代表单元的总自由度数，ndofn 代表节点自由度数。

（5）整型变量的中间三个字符通常为变量英文含义的简写。如 kdofe 中的 dof 代表自由度，nmats 中 mat 代表材料。

（6）浮点变量由相应英文含义简写得到，如果简写后的第一个字母代表隐含的整数，则在它之前加 v（代表数组）或 f（代表标量）。

在程序的研制过程中，我们还规定：

（7）内部函数和过程控制语句采用大写字母，变量一律采用小写字母。

（8）尽量避免采用无条件转移语句，少用条件转移语句。

（9）执行语句尽量少用语句标号，书写格式整齐，缩进有序。

12.2.3 主要数据模块

程序主要采用MODULE来传递数据，在文件Data.f90中包含了十多个这样的数据模块，这些模块不仅给出了变量和数组的定义，而且通过其包含的子程序完成所有可调数组的内存申请、变量和数组的初始化以及释放可调数组的内存空间。

CtrlData	分析程序运算过程中的所有控制参数。
MacroData	与宏命令相关的变量和数组，并负责从文件中读入宏命令集。
MeshData	与有限元离散模型相关的变量和数组。
ExtraMesh	有限元离散模型的附加信息。主要应用于改进的等参单元。
AutoMeshData	映射网格自动剖分时需要的所有信息。
CentData	质量块、弹簧、粘壶相关信息。
LoadData	载荷信息。
ElmtData	包含程序运行过程中单元库调用具体某个单元时需要传递的基本参数，包括单元节点编号、单元节点坐标、单元节点位移、速度和加速度等。
GlobData	有限元离散系统的状态信息。如系统的节点位移、速度、加速度等。
SolvData	程序分析过程中需要的工作变量和数组。
FrontData	波前法求解过程中需要的工作参数和数组。
FactorData	计算载荷参数（因子）时需要的变量和数组。
EarthQuake	保存地震加速度所需要的变量和数组。负责读入地震加速度时程曲线，并完成幅值调整。
FileData	输入输出文件名数据模块，负责读入输入、输出文件名，或设定缺省的数据文件名。

12.2.4 主要变量

有限元分析涉及众多方面的信息，分析程序往往包含大量的变量和数组，VFEAP也不例外。在翻阅源程序文本和再开发之前，熟悉这些变量是十分必要的。由于变量太多，表12.1和12.2只列出了其中一些常用变量和常用数组，并给予简单的说明。至于其他变量，请读者阅读附录中Data.f90文本。

表12.1 VFEAP中的主要变量

变量	所在模块	说明	备注
alpha	CtrlData	Newmark 法中的系数	缺省值为 0.5
belta	CtrlData	Newmark 法中的系数	缺省值为 0.25
fctor	CtrlData	载荷因子	
kload	CtrlData	当前分析工况序号	
kmacr	MacroData	当前执行的宏命令序号	
mcmds	MacroData	命令内部编号的最大值	参见 Macro.set 文件
mdofe	CtrlData	单元最大自由度数目	
mdofn	CtrlData	节点最大自由度数目	
mdofw	FrontData	波阵中最大自由度数	
mmacr	MacroData	最大宏命令数目	
mnode	CtrlData	单元最大的节点数	
mstfw	FrontData	波前刚度阵元素数	
mswth	CtrlData	单元选项开关最大值	设定为 100
nbufs	CtrlData	求解缓存区大小	
ndamp	CtrlData	粘壶数	
ndime	CtrlData	结构维数	范围：1-3
ndofe	ElmtData	单元自由度数	
ndofn	ElmtData	单元自由度数	
ndofs	CtrlData	系统自由度总数	
ndsks	CtrlData	求解外缓存区大小	
nelem	CtrlData	系统单元总数	
nerrc	CtrlData	运行错误序号	
nhstr	CtrlData	单元历史量数	
nincc	CtrlData	求解方法控制参数	
nlnkc	CtrlData	相同坐标节点自动连接标记	0：禁止；1：允许
nload	CtrlData	载荷工况总数	
nmacr	MacroData	当前算法宏命令数	
nmats	CtrlData	材料类型数	
nmode	CtrlData	模态数	
nmssp	CtrlData	质量块数	
nnode	ElmtData	单元节点数	
nnodg	CtrlData	定义线分布载荷的节点数	范围：2-3
nnodp	CtrlData	定义面分布载荷的节点数	范围：4-8
npoin	CtrlData	系统节点总数	
nprnc	CtrlData	模型数据数检输出标记	0：禁止；1：允许
nprop	CtrlData	材料性质参数数	
nrcrd	FrontData	外存记录数	
nslnt	CtrlData	斜约束总数	
nstfp	CtrlData	弹簧数	
ntypc	CtrlData	问题类型控制参数	参考数据文件输入格式
sdatf	FileData	模型数据文件名	不超过 256 个字符
sdisf	FileData	节点位移数据文件名	不超过 256 个字符
sdsvf	FileData	节点位移保存文件名	不超过 256 个字符

shisf	FileData	历史量数据文件名	不超过 256 个字符
sinif	FileData	初始条件文件名	不超过 256 个字符
smacf	FileData	算法文件名	不超过 256 个字符
smshf	FileData	后处理模型文件名	不超过 256 个字符
soutf	FileData	输出数据文件名	不超过 256 个字符
spref	FileData	单元预应力文件名	不超过 256 个字符
ssavf	FileData	运算过程状态保存文件名	不超过 256 个字符
sstaf	FileData	单元状态文件名	不超过 256 个字符
sstrf	FileData	节点应力文件名	不超过 256 个字符
strnf	FileData	节点应变文件名	不超过 256 个字符
theta	CtrlData	Wilson 法中的系数	缺省值为 1.4
timec	CtrlData	当前时刻	
title	CtrlData	分析问题的标题	不超过 70 个字符
tstep	CtrlData	时间步长	缺省值为 1

表12.2　VFEAP中的主要数组

变量	数据模块	说　明	备　注
accel(ndofs,nload)	GlobData	当前迭代步系统节点加速度	
accen(mdofe)	ElmtData	当前迭代步单元节点加速度	
acceq(naceq)	EarthQuake	地震加速度序列	
accet(mdofe)	ElmtData	上一时间步单元节点加速度	
acctm(ndofs,nload)	GlobData	上一时间步系统节点加速度	
aitkn(ndofs)	ExecData	Aitken 非线性加速因子	
ampeq(mdofn)	EarthQuake	地震加速度调幅量	
axisw(ndime,ndime)	MeshData	波阵优化主方向	
basew(ndime)	MeshData	波阵优化基准点	
buffs(nbufs)	ExecData	求解缓存区	
coord(ndime,npoin)	MeshData	节点坐标	
coren(ndime,mnode)	ElmtData	当前单元的节点坐标	初始构型
diagd(ndofs)	ExecData	总阻尼阵对角元素	
diagk(ndofs)	ExecData	总刚阵对角元素	
diagm(ndofs)	ExecData	总质量阵对角元素	
dispe(mdofe)	ElmtData	当前迭代步单元节点位移	
disps(ndofs,nload)	GlobData	当前迭代步系统节点位移	
dispt(mdofe)	ElmtData	上一时间步单元节点位移	
distm(ndofs,nload)	GlobData	上一时间步系统节点位移	
dspis(ndofs,nload)	GlobData	上一迭代步系统节点位移	
fbcks(nbcks)	FrontData	回代信息	
force(mdofe)	ElmtData	当前单元等效节点载荷	
forcs(ndofs)	SolvData	当前时间步外力等效节点力	
frceq(mdofn)	EarthQuake	自由度调频	暂缺
frctm(ndofs)	SolvData	上一时间步外力等效节点力	
freqs(mmode)	GlobData	结构固有频率或临界载荷	
fstrs(ndofs)	SolvData	当前时间步内力等效节点力	取负值

fstrt(ndofs)	SolvData	上一时间步内力等效节点力	取负值
fwork(ndofs)	SolvData	工作数组	
histr(nhstr,nelem)	GlobData	单元历史量	
lbcks(4,npoin)	FrontData	节点回代信息	
ldecc(3)	CtrlData	总体矩阵分解信息	
ldmpp(2,mdmpp)	MeshData	粘壶所在节点和自由度序号	
ldofs(npoin)	GlobData	节点自由度数	
ledgf(mnodg,medgf)	LoadData	线载荷整型参数	节点和载荷因子
lfixs(ndofn,npoin)	MeshData	节点自由度约束信息	0：自由；1：约束
llnks(npoin)	MeshData	节点连接信息	
lmacr(5, mmacr)	MacroData	当前算法宏命令参数	
lmats(nelem)	MeshData	单元材料序号	
lmssp(2,mmssp)	MeshData	质量块信息	节点和自由度序号
lnode(mnode)	ElmtData	当前单元的节点序号	
lnodf(2,mnodf)	LoadData	集中荷载整型参数	节点和载荷因子
lnods(mnode,nelem)	MeshData	构成单元的节点序号	
loads(3,nload)	LoadData	各工况点、线、面载荷数	
lopts(nelem)	FrontData	优化后的单元顺序	
lplnf(ndofp,mplnf)	LoadData	面载荷整型参数	节点和载荷因子
lpnte(npoin)	FrontData	围绕节点的单元数	
lpntw(npoin)	FrontData	波阵中节点序号	
lposw(npoin)	FrontData	节点在波阵中的位置	
lprps(nmatp,nmats)	MeshData	材料整型性质参数	
lrcrd(3,nrcrd)	FrontData	记录信息	
lsavc(3)	CtrlData	总体矩阵保存信息	
lslnt(2,nslnt)	MeshData	斜约束节点和自由度序号	
lstat(npoin)	GlobData	单元状态量	
lstfp(2,mstfp)	MeshData	集中弹簧所在节点和自由度序号	
lwave(mnode)	FrontData	单元节点在波阵中的位置	
macro(mcmds)	MacroData	命令集	
mrems(mcmds)	MacroData	宏命令的说明	
pmacr(5, mmacr)	MacroData	当前算法宏命令参数	
promp(4,50)	FactorData	载荷系数的模型参数	
props(nprop, nmats)	MeshData	材料性质参数	
react(ndofs)	GlobData	系统节点反力	
shift(10)	CtrlData	移频系数	
shmod(ndofs,mmode)	GlobData	结构固有模态或失稳模态	
statf(ndofs)	SolvData	约束处理时附加右端项	
stife(mdofe,mdofe)	ElmtData	单元元素矩阵	
stifw(mstfw)	FrontData	保存波阵的内缓存	
strns(7,npoin)	GlobData	系统节点应变	
strss (7,npoin)	GlobData	节点应力分量、应力加权	
tempe(mnode)	ElmtData	当前单元的节点温度	
temps(npoin)	MeshData	系统节点温度	

timeq(naceq)	EarthQuake	地震加速度时间序列	
twork(ndofs)	SolvData	工作数组	
vdmpp(mdmpp)	MeshData	粘壶阻尼系数	
vedgf(ndofg,medgf)	LoadData	线分布载荷各点分布集度	
velen(mdofe)	ElmtData	当前迭代步单元节点速度	
velet(mdofe)	ElmtData	上一时间步单元节点速度	
veloc(ndofs,nload)	GlobData	当前迭代步系统节点速度	
veltm(ndofs,nload)	GlobData	上一时间步系统节点速度	
vfixs(ndofn,npoin)	MeshData	给定的位移值	
vmssp(mmssp)	MeshData	作用在节点上的集中质量	
vnodf(mdofn,mnodf)	LoadData	集中载荷信息	载荷分量
vplnf(ndofp,mplnf)	LoadData	面分布载荷各节点分布集度	
vslnt(mdofn,nslnt)	MeshData	斜约束自由度的合成分量	
vslvs(ndofs)	SolvData	方程组的右端项	
vstfp(mstfp)	MeshData	弹簧弹性系数	

12.3　VFEAP 的宏命令和算法组织[1]

宏命令可以有效地组织各种算法，控制计算过程，因此具有使用方便、灵活的特点。VFEAP中宏命令的格式为

COMMAND　p1　p2　p3

其中，COMMAND为宏命令，p1、p2、p3是宏命令参数，为整数或实数，根据宏命令的要求输入，为可选项。所有已开发的宏命令保存在文件Macro.set中。宏命令不区分大小写字母，宏命令和参数之间用空格分开，参数和参数之间用空格或逗号隔开。

12.3.1　宏命令

目前Macro.set中有宏命令60多个，表12.3列出了其中常用的一些算法宏命令及其说明。

表12.3　VFEAP中的主要宏命令

宏命令	功能	参数一	参数二	参数三
accel	计算初始加速度			
alpha	为Newmark法中的参数赋值	Alpha值		
belta	为Newmark法中的参数赋值	Belta值		
cmass	计算质量阵			
construct	构建单元	起始单元	终止单元	单元步长
crank	Crank法求解			
damp	计算阻尼阵			
destruct	拆除单元	起始单元	终止单元	单元步长
echofctor	显示载荷因子			
Echoon	显示计算过程			
echotime	显示当前时间			
eigenpair	求解广义特征值	容差(2)		

endloop	循环体结束			
fix	设置约束	节点	自由度	
force	计算等效节点力			
free	释放约束	节点	自由度	
geometry	计算几何刚度阵			
initial	输入初始条件			
loadcase	指定当前工况	工况序号		
loop	开始循环	起始值	终止值	
newmark	Newmark法求解响应			
nextload	指向下一工况			
nodestrs	计算节点应力			
pause	分析暂停			
phasedraw	显示计算过程曲线	类型(5)	节点	自由度
plotdisp	输出节点位移到dsp文件			
plotmesh	输出模型几何到msh文件			
plotstate	输出单元状态到sta文件			
plotstrs	输出节点应力到str文件			
printacc	输出节点加速度到out文件	起始节点	终止节点	节点步长
printdisp	输出节点位移到out文件	起始节点	终止节点	节点步长
printstat	输出单元状态到out文件			
printvel	输出节点速度到out文件	起始节点	终止节点	节点步长
react	计算节点反力			
setmat	指定材料	起始单元	终止单元	单元步长
shift	移频	移频量(4)		
solve	求解			
stiffen	计算刚度阵	整数(3)		
stress	计算单元应力	起始单元	终止单元	单元步长
theta	指定Wilson法参数	参数		
timedraw	显示时程曲线	类型(5)	节点	自由度
timestep	设定时间步长	步长		
tolen(1)	容差判断	容差(2)		
trace(6)	跟踪物理量	类型(5)	节点	自由度
update	时间步更新			
wilson	Wilson法求解响应			

(1) tolen必须位于一个LOOP循环内，当计算结果满足容差时，跳出该层循环。

(2) 相对容差，缺省值为10^{-5}，大于等于1时代表负指数。如6代表10^{-6}，小于1时该参数就是相对容差。

(3) 缺省值为0，代表计算切线刚度阵的同时计算内力等效节点力矢量，如果不等于零，只计算内力等效节点力矢量。

(4) 自由振动分析时为圆频率。

(5) 类型取值0~3，0或1代表位移，2代表速度，3代表加速度。

(6) 结果输出在文件H#××××××※.dat中，其中，#为D、V或A，分表代表位移、速度和加速度；×××××代表节点序号，※代表自由度序号。如HD000021.dat为节点2自由度1的时程数据文件。

12.3.2　算法组织

每一个问题的求解都需要一组以"macro"开头、"endmacro"结尾的宏命令来控制计算过程。这组宏命令保存在mac文件中。利用宏命令可以组织多种算法，以下列出几种常用算法，供读者参考。

一、 静力有限元问题

(1) 线性问题

```
MACRO
    stiffen          //计算整体刚度矩阵
    force            //计算右端项
    solve            //求解线性方程组
    nodestrs         //计算节点应力(根据需要选择)
    plotmesh         //输出有限元模型数据(根据需要选择)
    plotdisp         //输出有限元变形数据(根据需要选择)
    plotstrs         //输出有限元应力数据(根据需要选择)
ENDMACRO
```

(2) 非线性问题 Newton-Raphson 迭代法

```
MACRO
    LOOP 1 10        //最大迭代次数
        stiffen      //计算整体刚度矩阵
        force        //计算右端项
        solve        //求解线性方程组
        tolen 5      //计算位移矢量精度为10^-5
    ENDLOOP
    nodestrs         //计算节点应力(根据需要选择)
    plotmesh         //输出有限元模型数据(根据需要选择)
    plotdisp         //输出有限元变形数据(根据需要选择)
    plotstrs         //输出有限元应力数据(根据需要选择)
ENDMACRO
```

(3) 非线性问题的修正 Newton-Raphson 迭代法

```
MACRO
    LOOP 1 10            //10 为最大变刚度迭代次数
        stiffen          //计算整体刚度矩阵
        LOOP 1 5             //5 为最大等刚度迭代次数
            stiffen 1        //计算内力等效节点力
            force            //计算右端项
            solve            //求解线性方程组
            tolen 5          //计算位移矢量精度为10^-5
        ENDLOOP
        tolen 5          //计算位移矢量精度为10^-5
    ENDLOOP
```

```
        nodestrs                //计算节点应力(根据需要选择)
        plotmesh                //输出有限元模型数据(根据需要选择)
        plotdisp                //输出有限元变形数据(根据需要选择)
        plotstrs                //输出有限元应力数据(根据需要选择)
    ENDMACRO
```

(4) 非线性问题的载荷增量法（结合修正的 Newton-Raphson 迭代法)

```
    MACRO
        timestep 0.1        //时间步长为 0.1
        LOOP 1 10           //10 为载荷步数
            update          //更新时间步(载荷步)
            stiffen         //计算整体刚度矩阵
            LOOP 1 5            //5 为最大等刚度迭代次数
                stiffen 1       //计算内力等效节点力
                force           //计算右端项
                solve           //求解线性方程组
                tolen 5         //计算位移矢量精度为10^-5
            ENDLOOP
            tolen 5         //计算位移矢量精度为10^-5
        ENDLOOP
        nodestrs            //计算节点应力(根据需要选择)
        plotmesh            //输出有限元模型数据(根据需要选择)
        plotdisp            //输出有限元变形数据(根据需要选择)
        plotstrs            //输出有限元应力数据(根据需要选择)
    ENDMACRO
```

二、自由振动问题

```
    MACRO
        cmass               //计算质量阵
        stiffen             //计算刚度阵
        eigenpair   5       //计算特征对(迭代精度为10^-5)
        plotmesh            //输出有限元模型数据(根据需要选择)
        plotdisp            //输出固有模态变形数据(根据需要选择)
    ENDMACRO
```

三、 临界荷载问题

```
    MACRO
        stiffen             //计算刚度阵
        force               //计算右端项
        solve               //求解线性方程组
        geometry            //计算几何刚度阵
        stiffen             //计算刚度阵
        eigenpair   5       //计算特征对(迭代精度为10^-5)
```

```
        plotmesh          //输出有限元模型数据(根据需要选择)
        plotdisp          //输出固有模态变形数据(根据需要选择)
    ENDMACRO
```

四、瞬态响应问题(以 Newmark 法为例)

(1) 无阻尼线性结构

```
    MACRO
        timestep 0.01     //时间步长为 0.01
        initial           //输入初始条件
        accel             //计算初始加速度
        cmass             //计算质量阵
        stiffen           //计算刚度阵
        newmark           //Newmark 法计算等效刚度阵
        LOOP 1 200        //时间步为 200
            update        //更新时间步
            force         //计算右端项
            solve         //求解线性方程组、速度、加速度
            trace 1 5 2   //输出 5 节点 2 自由度的位移时程
        ENDLOOP
    ENDMACRO
```

(2) 无阻尼非线性结构(结合修正的 Newton-Raphson 迭代法)

```
    MACRO
        timestep 1 0.04   //时间步长为 0.04
        initial           //输入初始条件
        accel             //计算初始加速度
        cmass             //计算质量阵
        LOOP 1 200        //时间步为 200
            update        //更新时间步
            stiffen       //计算刚度阵
            newmark       //Newmark 法计算等效刚度阵
            LOOP 1 20     //20 为最大迭代步数
                force     //计算右端项
                solve     //求解线性方程组、速度、加速度
                tolen 5   //计算位移矢量精度为10^-5
            ENDLOOP
            trace 1 5 2   //输出 5 节点 2 自由度的位移时程
        ENDLOOP
    ENDMACRO
```

12.4 主要输入输出文件及格式

运行VFEAP一般需要三个不同的数据文件：有限元模型数据文件(扩展名为DAT，简称DAT文件)、算法宏命令文件（扩展名为MAC，简称为MAC文件）和初始条件数据文件（扩展名为INI，简称INI文件）。DAT文件和MAC文件是VFEAP进行分析时必不可少的，而INI文件只有在与时间相关、且具有非零初始条件的问题中才需要。

12.4.1 有限元模型数据文件

有限元模型数据文件是一个格式输入文件，文件格式及输入变量的含义按读入的先后次序依次说明如下。

一、标题

程序允许在标题行输入不超过70个字符的字符串作为分析工作的标题或说明。标题共一行，输入变量和格式为：

title *(A70)*

二、主控参数

主控参数共有三行。

第一行：ntypc, npoin, nelem, ndime, mdofn, nmats, nload, nslnt*(8I5)*

ntypc		问题的类型
	== 0	静力分析
	== 1	无阻尼结构自由振动分析
	== 2	无阻尼结构瞬态响应分析
	== 3	结构临界荷载分析
	== 4	有阻尼结构自由振动分析
	== 5	有阻尼结构瞬态响应分析
npoin		系统节点总数
nelem		系统单元总数
ndime		问题维数
mdofn		节点最大自由度数
nmats		材料类型数
nload		工况数
nslnt		斜约束自由度数

第二行：mnode, nnodp, nnodg, nprop, mnodf, medgf, mplnf, nprnc *(8I5)*

mnode	单元最大节点数
nnodp	定义面分布载荷需要的节点数
nnodg	定义线分布载荷需要的节点数
nprop	材料性质参数数
mnodf	集中载荷总数（包括所有工况）
medgf	线分布载荷总数（包括所有工况）
mplnf	面分布载荷总数（包括所有工况）

nprnc　　输入数据输出控制

= 0　禁止；

= 1　允许；

第三行：nmssp, nstfp, ndmpp, nmode, nlnkc *(5I5)*

nmssp　　质量块数（按自由度计算）

nstfp　　弹簧数（按自由度计算）

ndmpp　　粘壶数（按自由度计算）

nmode　　模态数

nlnkc　　具有相同坐标的节点是否自动连接为一个节点

== 0　禁止自动连接；

== 1　允许自动连接；

三、节点坐标和温度

节点坐标和温度输入的行数取决于离散系统的规模和节点编号及节点坐标的规律性。每一行输入一个节点的坐标，每两行之间可以自动生成节点坐标。每条记录的格式如下，

ipoin, ictyp, igene, corpi, tempi *(3I5，4F15.5)*

ipoin　　当前输入的节点序号

ictyp　　坐标类型

== 0 直角坐标

== 1 极(柱)坐标

== 2 球坐标

igene　　坐标自动生成的节点步长

corpi　　当前节点的坐标

tempi　　当前节点温度

节点坐标输入时注意以下两点：

（1）当前输入的节点编号为0（或空行）或大于节点总数时，表示节点坐标输入结束。否则，程序将继续读入节点坐标。

（2）坐标自动生成的规则是，从上一记录输入的节点到当前记录输入的节点，按上一记录输入的节点步长自动生成中间节点的坐标和温度，坐标和温度均匀插值。

（3）只有坐标类型相同的两节点间才可自动生成节点坐标。

四、节点连接

程序除可以自动地将坐标相同的两个节点连接为一个节点外，也可在此强制若干节点具有相同的位移。输入格式为

(lwork(i), i = 1, 10) *(10I5)*

一行可最多输入10个节点编号，这些节点将被程序强制具有相同的位移。如果一行中的所有节点编号全为零（或空行），表示连接信息输入结束，否则程序将继续读入连接信息。

五、单元定义

单元定义的输入和节点坐标输入类似。当单元最大节点数少于10个时，定义一个单元只需要一行，否则需要两行或更多行。格式如下：

ielem, imats, igene, lnode*(13I5 / 15x，10I5)*

ielem　　当前输入的单元编号

imate	当前单元对应的材料序号
igene	单元定义自动生成的单元步长
lnode	当前单元的节点定义(共mnode个整数)

单元定义要注意以下几点，

（1）单元序号为0（或空行）或大于单元总数时，表示单元定义输入结束，否则程序将继续读入单元定义。

（2）单元自动生成的规则和节点坐标相似，单元各节点序号均匀插值，不恰当的节点编号可能导致程序运行错误。

（3）只有单元类型相同、材料序号相同的两个单元间才可自动生成。

六、节点约束

节点约束信息是按自由度输入的，格式如下，

ipoin, idofn, itype, vcord, value***(3I5，2F15.5)***

ipoin	约束节点序号
idofn	约束自由度号
itype	约束位移随时间变化的类型序号
vcord	坐标面的坐标值
value	给定位移值

输入约束信息时特别注意以下几点，

（1）当节点序号小于零时，表示对应的某坐标面的所有节点都具有该约束，-1、-2和-3分别代表x、y和z坐标面。如记录

-1　1　0　1.00000　0.00000

表示x坐标为1.0的坐标面上所有节点的第一个自由度均为零位移。

（2）节点号为0（或空行）表示约束信息输入结束，否则程序将继续读入约束信息。

七、材料信息

材料信息分两部分，第一部分对所有单元来说都是相同的，第二部分则需根据单元具体确定输入量。第一部分占一行，一般只需输入两个参数，格式为

imate, ietyp***(2I5)***

其中，

imate	当前输入材料的序号
ietyp	当前材料类型对应的单元类型序号

八、载荷信息

对每一载荷工况依次输入以下信息，直到所有工况输入完毕。

（1）工况基本控制信息

nnodf, nedgf, nplnf ***(3I5)***

其中，

nnodf	当前工况的集中载荷数
nedgf	当前工况的线分布载荷数
nplnf	当前工况的面分布载荷数

（2）集中载荷信息

集中载荷信息共有nnodf条记录，无集中载荷时无需填写，每条记录的格式如下，

ipoin, itype, (fwork(idofn), idofn = 1, mdofn) ***(2I5, 4F15.5 / (10X, 4F15.5))***

其中，

ipoin	集中载荷作用的节点序号
itype	荷载随时间变化的类型序号
fwork	作用在节点上的荷载分量

（3）线分布载荷信息

线分布载荷信息共nedgf个，无线分布载荷时无需填写。线分布载荷按作用节点依次输入，规定先角节点后边中节点。每个节点的输入格式为

ipoin, itype, (fwork(idofn), idofn = 1, mdofn) ***(2I5, 4F15.5 / (10X, 4F15.5))***

其中，

ipoin	线分布载荷作用的节点序号
itype	荷载随时间变化的类型序号
fwork	作用在节点上的荷载线分布集度

（4）面分布载荷信息

面分布载荷信息共nplnf个，没有面分布荷载时无需填写。面分布载荷的输入方法和格式与线分布荷载完全相同，注意不同的荷载分布集度的单位，这里不再重复。

九、质量块信息

共nmssp条记录，无质量块时无需填写。质量块信息的输入格式为

ipoin, idofn, fmass ***(2I5,F15.5)***

其中，

ipoin	质量块作用的节点序号
idofn	质量块作用的自由度序号
fmass	质量块质量

十、弹簧信息

共nstfp条记录，无弹簧时无需填写。弹簧信息的输入格式为

ipoin, idofn, fstif***(2I5,F15.5)***

其中，

ipoin	弹簧作用的节点序号
idofn	弹簧作用的自由度序号
fstif	弹簧弹性系数

十一、粘壶信息

共ndmpp条记录，无粘壶时无需填写。粘壶信息的输入格式为

ipoin, idofn, fdamp***(2I5,F15.5)***

其中，

ipoin	粘壶作用的节点序号
idofn	粘壶作用的自由度序号
fdamp	粘壶阻尼系数

十二、斜约束信息

定义斜约束自由度，共nslnt条记录，每条记录的格式为

ipoin, idofn, (vslnt(jdofn, islnt), jdofn = 1, mdofn) ***(2I5, 4F15.5 / (10X, 4F15.5))***

ipoin 斜约束自由度所在节点序号；

idofn 斜约束自由度序号；

vslnt 斜约束自由度与正常自由度间的转化关系。

与约束信息相同，ipoin为负值时代表某一坐标面的所有节点均具有斜约束自由度。

十三、波阵带宽优化参考坐标系

输入参考系的原点和三个坐标轴的单位方向矢量。中心点和方向矢量输入时共占四行，每行的格式为***(3F15.5)***

12.4.2 宏命令文件

宏命令文件采用自由格式，以MACRO开头，以ENDMACRO结尾，其间的所有命令视为有效，其他语句均被视为说明语句。

12.4.3 初始条件数据文件

对于与时间相关的问题，在初始条件数据文件中依次输入初始位移和初始速度，两者格式相同，都是按自由度输入的。每条记录的格式均为***(2I5,F15.5)***，输入的物理量为：节点序号、自由度序号、初始位移或初始速度值。第一条0节点序号记录（或空行）代表初始位移输入完毕，第二条0节点序号记录（或空行）代表初始速度输入完毕。

12.5 宏命令和单元库的扩展

由于采用了模块化设计，且模块与模块之间相互干扰比较小，因此，能很方便地实现程序功能扩展。以下介绍VFEAP中新的宏命令和新单元的添加方法。

12.5.1 宏命令扩展

虽然VFEAP已包含了为数不少的宏命令，但有时读者仍然需要添加自己的宏命令，以实现某些特殊功能。在VFEAP中添加宏命令需要完成以下三个步骤。

(1) 确定宏命令、宏命令编号和宏命令参数，在文件MACRO.SET中登记。注意用户自己添加的宏命令编号不得与已有的宏命令编号相同，也不得随意修改其他宏命令的编号。

(2) 在文件Execute.f90中对应位置添加实现该宏命令的分析语句。由于宏命令众多，在一个子程序中完成所有宏命令的处理过程将使得该子程序过于庞大，不易管理。因此，子程序Execute根据宏命令的序号将处理过程划分到六个子处理模块中。宏命令编号和处理子程序间的对应关系见表12.4。

表12.4 宏命令编号与处理子程序

宏命令编号	处理子程序	宏命令编号	处理子程序
1～20	Exec020	21～40	Exec040
41～60	Exec060	61～80	Exec080
81～120	Exec120	121～140	Exec140

注意，在处理子程序中，kmacr代表当前正在处理的宏命令序号，数组lmacr和pmacr中保

存了宏命令的参数。其中，lmacr(2,kmacr)指向下一条待处理的宏命令。为确保当前命令执行完毕后自动执行下一条宏命令，一般要求在当前宏命令处理完毕后加如下语句：

```
kmacr = lmacr( 2, kmacr )
```

否则程序会陷入死循环。

(3) 菜单选择Build->Build，重新编译连接。

12.5.2　添加单元

在VFEAP中添加新的单元是比较方便的，使用者只要按要求开发新的单元，并在ElmLib.f90中注册调用后即可。具体步骤如下：

一、 按规定格式编制新单元

VFEAP中的单元应符合以下规定。

(1) 单元子程序的命名方法。单元子程序采用统一的命名方法，即Elmt***，其中***为该单元对应的序号，如Elmt200即代表200号单元。

(2) 单元子程序的结构。子程序的结构如下：

```
SUBROUTINE Elmt***( ielem, imats, iswth, iuses ) //子程序原型
USE CtrlData                                     //引用控制数据
USE MeshData                                     //引用网格数据
USE ElmtData                                     //引用单元信息
USE FactorData                                   //引用载荷因子
USE GlobData                                     //引用状态数据
IMPLICIT DOUBLE PRECISION ( a-h , o-z )          //变量隐含说明
DIMENSION ……                                    //数组说明语句
  ……                                             //数据初始化
IF( iswth .EQ. 0 ) THEN                          //依次完成功能
   ……
ELSE IF( iswth .EQ. 1 ) THEN
  ……
END IF
RETURN
END
```

子程序的虚参说明如下：

ielem　　当前单元序号

imats　　当前单元对应的材料序号

iswth　　功能选项

== 0　读入材料参数、确定单元历史量数和求解方法

== 1　计算单元刚度矩阵和负的内力等效节点力

== 2　计算单元质量阵

== 3　计算单元阻尼矩阵

== 4　计算单元的几何刚度矩阵

== 5　计算单元负的内力等效节点力

```
==6 计算单元体积力等效节点力
==7 计算单元应力
==8 确定单元基本参数
==9 输出单元后处理信息
==10 求反力
==11 计算单元的历史量
```

二、在文件ELMLIB.F90的ElmOpt模块中注册调用

在选择语句块中添加调用新单元的语句。假设新增加的单元为205号单元，则只需增加如下两条语句即可。

```
itype = IABS( lprps( 1, imats ) )                    //获得当前单元的单元类型
SELECT CASE( itype )                                 //根据单元类型选择单元模块
  CASE( 1 )
    CALL Elmt01( ielem, imats, iswth, iuses )        //调用单元1
  CASE( 2 )
    CALL Elmt02( ielem, imats, iswth, iuses )        //调用单元2
  CASE( 3 )
    CALL Elmt03( ielem, imats, iswth, iuses )        //调用单元3
  CASE( 4 )
    ⋮
  CASE( 205 )                                        //新单元调用
    CALL Elmt205( ielem, imats, iswth, iuses )
        ⋮
  CASE( 70 )
    CALL Elmt70( ielem, imats, iswth, iuses )        //调用单元70
  CASE DEFAULT                                       //调用了没有开发的单元类型
    nerrc = 24245
    WRITE( 12, 2000 ) itype
END SELECT
```

三、将新增加的单元插入工程文件并编译连接

具体步骤为：

（1）选择菜单Insert→Files into Project；

（2）在对话框中选择新开发的单元文件Elmt***.f90;

（3）选择菜单Build→Build VFEAP.exe。

也可以按以下步骤进行，

（1）选择菜单File→Open，在对话框中选择新开发单元文件Elmt***.f90。如果该文件已经打开，通过Window菜单确保该文件所在窗口为活动窗口(Active Window)；

（2）选择菜单Build→Compile Elmt***.f90，提示是否将该文件加入当前工程文件，选择“是”；

（3）选择菜单Build→Build VFEAP.exe。

完成以上步骤后，就可以开始进一步调试单元了。

12.6 运行及输入输出样本

下面简单介绍VFEAP软件的使用和输入输出样本。

12.6.1 软件运行

运行VFEAP需要以下几个文件，

MACRO.SET	宏命令集文件
IOFile.EXE	确定输入输出文件名的可执行文件
MemSet.EXE	确定内外存信息的可执行文件
VFEAP.EXE	有限元分析程序
VFEAPCom.DLL	与其他可执行文件间信息交换的动态连接库

VFEAP软件安装或调试完毕后，先运行MemSet设置内外存信息，如图12.1示。对于规模小的有限元系统，建议设置较小的交换内存，以免占用太多的计算机资源。编辑完输入数据文件，开始分析前，启动IOFile，为有限元分析程序VFEAP设置输入输出文件，如图12.2示。最后，启动VFEAP开始有限元分析。

VFEAP将在运行过程中输出多个文本文件，具体数量由算法宏命令确定。其中最主要输出文件的扩展名为OUT，一般简称为OUT文件，其中包含所有输入数据和运行过程信息。

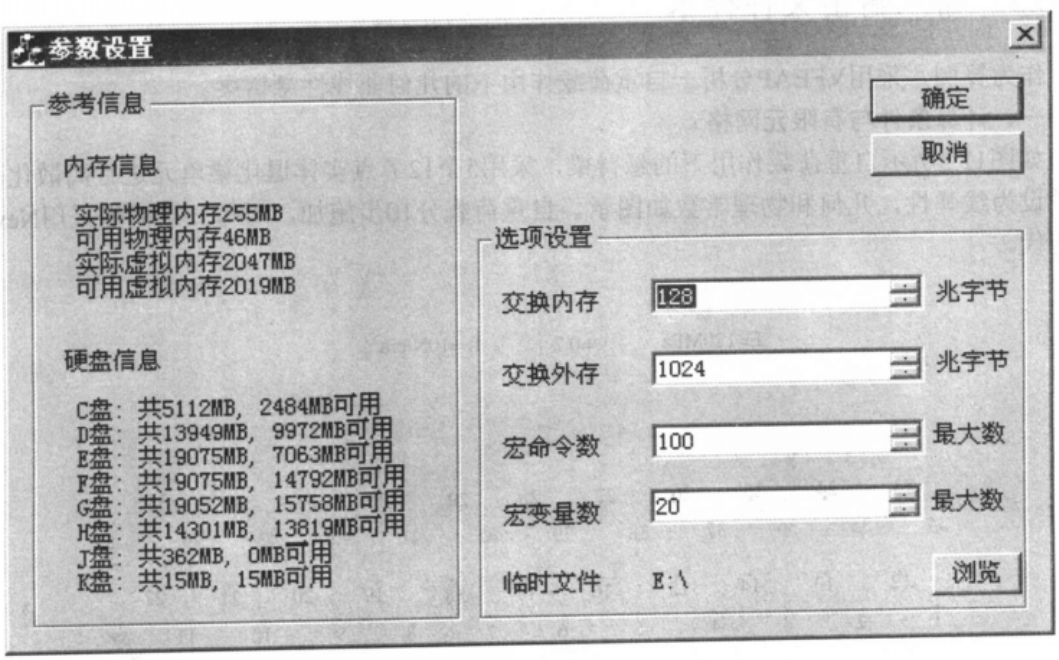


图12.1 运行MemSet的界面

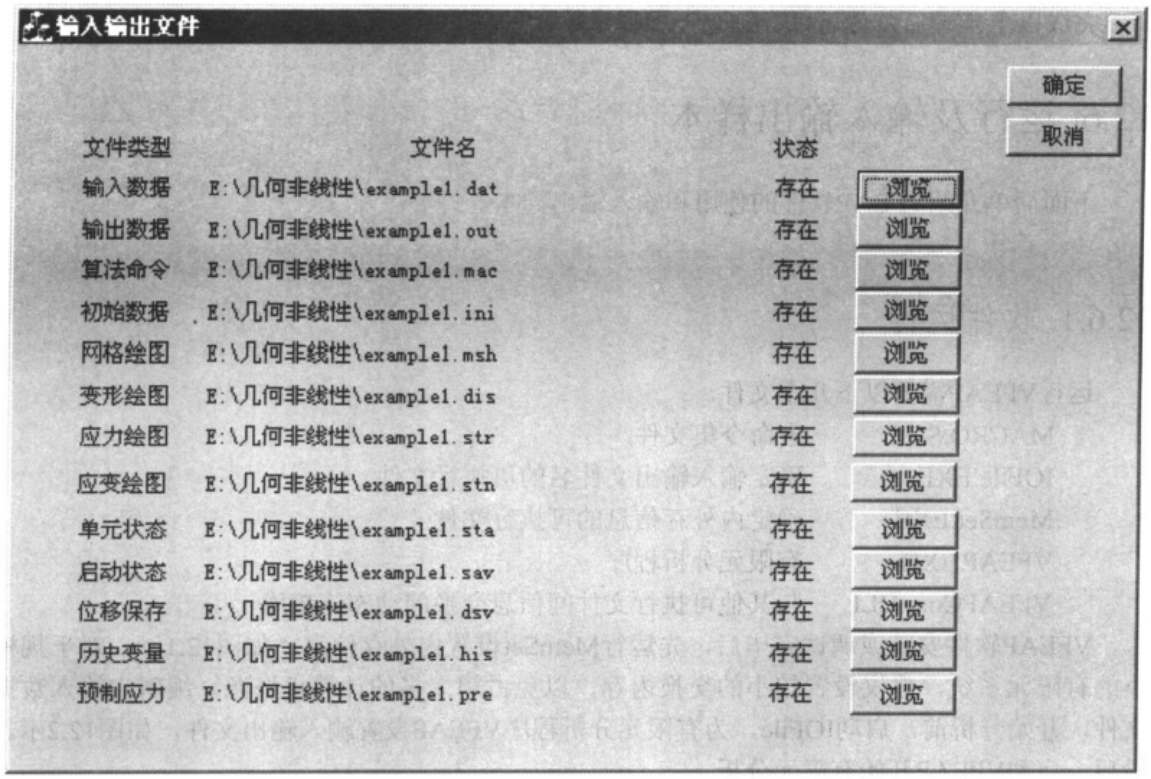


图12.2 运行IOFile的界面

12.6.2 输入输出文件样本

作为算例，采用VFEAP分析一自重荷载作用下的几何非线性悬臂梁。

一、计算条件与有限元网格

如图12.3所示自重荷载作用下的悬臂梁，采用5个12节点实体退化梁单元进行离散化。材料假设为线弹性，几何和物理常数如图示。自重荷载分10步施加，对每一载荷步采用Newton法迭代。

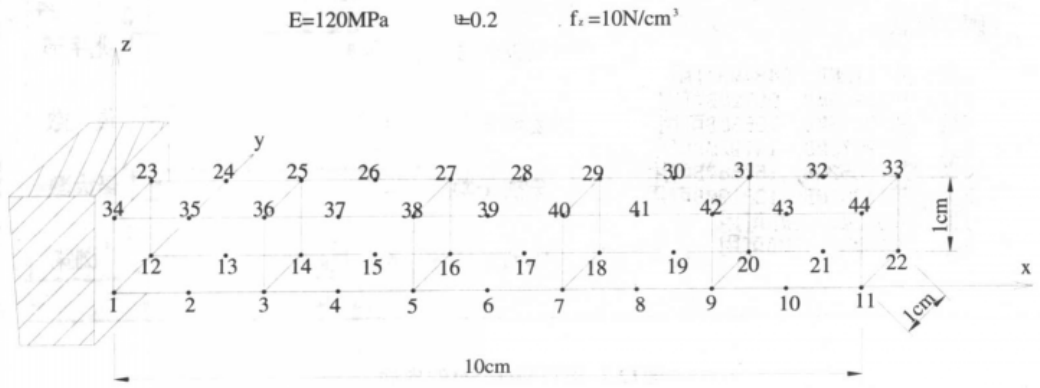


图12.3 自重荷载作用下的悬臂梁

二、输入数据文件样本

均布荷载作用下几何非线性悬臂梁

```
 0   44    5    3    3    3    1    0
20    8    3   40   10   10   10    1
 0    0    0    2    0    0
 1         1          0.00000          0.00000          0.00000
11                   10.00000          0.00000          0.00000
12         1          0.00000          1.00000          0.00000
22                   10.00000          1.00000          0.00000
23         1          0.00000          1.00000          1.00000
33                   10.00000          1.00000          1.00000
34         1          0.00000                           1.00000
44                   10.00000                           1.00000


 1    1    1    1    3   14   12   34   36   25   23    2   13
                24   35
 5    1         9   11   22   20   42   44   33   31   10   21
                32   43

-1    1
-1    2
-1    3

 1  300                          1
 1    1
 2    3
 2  300                          2
 3    8    8    3    2    2
     -1.00000         -1.00000         -1.00000
      1.00000         -1.00000         -1.00000
      1.00000          1.00000         -1.00000
     -1.00000          1.00000         -1.00000
     -1.00000         -1.00000          1.00000
      1.00000         -1.00000          1.00000
      1.00000          1.00000          1.00000
     -1.00000          1.00000          1.00000
 1    2    3    4    5    6    7    8
 3  300                          3
12000.00000          0.20000


    10.00000    1
```

三、算法宏命令文件

```
////////////////////////////////////
//      增量法+Newton迭代法求解
////////////////////////////////////

MACRO
    timestep 0.1
    LOOP 1 10
        update
        LOOP 1 50
          stiffen
          force
          solve
          tolen
        ENDLOOP
        trace 1 11 3
        timedraw 1 11 3
    ENDLOOP
ENDMACRO
```

四、输出数据文件样本

通用结构分析程序

==========

均布荷载作用下几何非线性悬臂梁

==========

问题类型	：静力分析		
节点总数	： 44	单元总数:	5
问题维数	： 3	节点自由度:	3
材料总数	： 3	载荷工况数:	1
单元节点数:	20	斜自由度数:	0
面载节点数:	8	线载节点数:	3
材性参数数:	40	集中载荷数:	10
线布载荷数:	10	面布载荷数:	10
集中质量数:	0	集中刚度数:	0
集中阻尼数:	0	最大模态数:	2

注意: @坐标相同或相近的节点将不被自动连接@

==== 节点坐标 ====

节点	x-坐标	y-坐标	z-坐标	节点温度	单元尺度
1	0.00000	0.00000	0.00000	0.00000	0.00000
2	1.00000	0.00000	0.00000	0.00000	0.00000
3	2.00000	0.00000	0.00000	0.00000	0.00000
4	3.00000	0.00000	0.00000	0.00000	0.00000
5	4.00000	0.00000	0.00000	0.00000	0.00000
6	5.00000	0.00000	0.00000	0.00000	0.00000
7	6.00000	0.00000	0.00000	0.00000	0.00000
8	7.00000	0.00000	0.00000	0.00000	0.00000
9	8.00000	0.00000	0.00000	0.00000	0.00000
10	9.00000	0.00000	0.00000	0.00000	0.00000
11	10.00000	0.00000	0.00000	0.00000	0.00000
12	0.00000	1.00000	0.00000	0.00000	0.00000
13	1.00000	1.00000	0.00000	0.00000	0.00000
14	2.00000	1.00000	0.00000	0.00000	0.00000
15	3.00000	1.00000	0.00000	0.00000	0.00000
16	4.00000	1.00000	0.00000	0.00000	0.00000
17	5.00000	1.00000	0.00000	0.00000	0.00000
18	6.00000	1.00000	0.00000	0.00000	0.00000
19	7.00000	1.00000	0.00000	0.00000	0.00000
20	8.00000	1.00000	0.00000	0.00000	0.00000
21	9.00000	1.00000	0.00000	0.00000	0.00000
22	10.00000	1.00000	0.00000	0.00000	0.00000
23	0.00000	1.00000	1.00000	0.00000	0.00000
24	1.00000	1.00000	1.00000	0.00000	0.00000
25	2.00000	1.00000	1.00000	0.00000	0.00000
26	3.00000	1.00000	1.00000	0.00000	0.00000
27	4.00000	1.00000	1.00000	0.00000	0.00000
28	5.00000	1.00000	1.00000	0.00000	0.00000
29	6.00000	1.00000	1.00000	0.00000	0.00000
30	7.00000	1.00000	1.00000	0.00000	0.00000
31	8.00000	1.00000	1.00000	0.00000	0.00000
32	9.00000	1.00000	1.00000	0.00000	0.00000
33	10.00000	1.00000	1.00000	0.00000	0.00000
34	0.00000	0.00000	1.00000	0.00000	0.00000
35	1.00000	0.00000	1.00000	0.00000	0.00000
36	2.00000	0.00000	1.00000	0.00000	0.00000
37	3.00000	0.00000	1.00000	0.00000	0.00000
38	4.00000	0.00000	1.00000	0.00000	0.00000
39	5.00000	0.00000	1.00000	0.00000	0.00000
40	6.00000	0.00000	1.00000	0.00000	0.00000
41	7.00000	0.00000	1.00000	0.00000	0.00000

42	8.00000	0.00000	1.00000	0.00000	0.00000
43	9.00000	0.00000	1.00000	0.00000	0.00000
44	10.00000	0.00000	1.00000	0.00000	0.00000

节点连接信息

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31

32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44

==== 单 元 信 息 ====

单元	材料	节点	节点	节点	节点	节点	节点	节点	节点	节点	节点	分段	分段	分段
1	1	1	3	14	12	34	36	25	23	2	13	0	0	0
		24	35	0	0	0	0	0	0	0	0			
2	1	3	5	16	14	36	38	27	25	4	15	0	0	0
		26	37	0	0	0	0	0	0	0	0			
3	1	5	7	18	16	38	40	29	27	6	17	0	0	0
		28	39	0	0	0	0	0	0	0	0			
4	1	7	9	20	18	40	42	31	29	8	19	0	0	0
		30	41	0	0	0	0	0	0	0	0			
5	1	9	11	22	20	42	44	33	31	10	21	0	0	0
		32	43	0	0	0	0	0	0	0	0			

==== 位 移 约 束 信 息 ====

节点	自由度		给定位移
1	1	0	0.00000
1	2	0	0.00000
1	3	0	0.00000
12	1	0	0.00000
12	2	0	0.00000
12	3	0	0.00000
23	1	0	0.00000
23	2	0	0.00000
23	3	0	0.00000
34	1	0	0.00000
34	2	0	0.00000

34 3 0 0.00000

==== 材 料 信 息 ====

材料类型序号 1
单元类型序号 300
材料作用因子 0
自由度序列表 0 0 0
分块数目: 1
几何特性: 几何非线性
各块特性: 序号 几何 材料
1 2 3

材料类型序号 2
单元类型序号 300
材料作用因子 0
自由度序列表 0 0 0
单元类型: 三维实体元(板壳元)分域几何
单元类型: 三维梁单元
x-高斯点: 3
y-高斯点: 2
z-高斯点: 2
节点坐标: 1 -1.00000 -1.00000 -1.00000
节点坐标: 2 1.00000 -1.00000 -1.00000
节点坐标: 3 1.00000 1.00000 -1.00000
节点坐标: 4 -1.00000 1.00000 -1.00000
节点坐标: 5 -1.00000 -1.00000 1.00000
节点坐标: 6 1.00000 -1.00000 1.00000
节点坐标: 7 1.00000 1.00000 1.00000
节点坐标: 8 -1.00000 1.00000 1.00000
分块定义: 1 2 3 4 5 6 7 8

材料类型序号 3
单元类型序号 300
材料作用因子 0
自由度序列表 0 0 0
单元类型: 三维实体元(板壳元)分层材料

单元材料:	线弹性材料
弹性模量:	0.12000E+05
泊 松 比:	0.20000
质量密度:	0.00000E+00
x-体积力:	0.00000E+00
y-体积力:	0.00000E+00
z-体积力:	0.10000E+02
动载类型:	1

载荷工况　1

优化中心和优化轴线信息

优化中心:	0.00000	0.00000	0.00000
第1轴线:	1.00000	0.00000	0.00000
第2轴线:	0.00000	1.00000	0.00000
第3轴线:	0.00000	0.00000	1.00000

波阵优化信息

波阵最大带宽　12(节点)

优化单元顺序

1　2　3　4　5

节点在波阵中的信息

当步波宽	总体节点号	波阵位置	
12	1	1	1
11	12	4	1
10	34	5	1
9	23	8	9
8	2	9	9
7	13	10	17
6	24	11	17
5	35	12	25
12	3	2	25
11	14	3	33

10	36	6	33
9	25	7	1
8	4	9	1
7	15	10	1
6	26	11	9
5	37	12	9
12	5	1	17
11	16	4	17
10	38	5	25
9	27	8	25
8	6	9	33
7	17	10	33
6	28	11	1
5	39	12	1
12	7	2	1
11	18	3	9
10	40	6	9
9	29	7	17
8	8	9	17
7	19	10	25
6	30	11	25
5	41	12	33
12	9	1	33
11	11	2	1
10	22	3	1
9	20	4	1
8	42	5	9
7	44	6	9
6	33	7	17
5	31	8	17
4	10	9	25
3	21	10	25
2	32	11	33
1	43	12	33

一致矩阵的存储信息

刚度阵存于内存，首址 1

外存记录信息

一致阵外存记录数 1

记录号 起始节点 终止节点 有效长度

1　1　44　2886

===== 宏 命 令 =====

宏命令	参数一	参数二	参数三	参数四	参数五
timestep	0.10000	0	0	0	0
loop	1	10	1	0	0
update	0	0	0	0	0
loop	1	50	1	0	0
stiffen	0	0	0	0	0
force	0	0	0	0	0
solve	0	0	0	0	0
tolen	0	0	0	0	0
endloop	0	0	0	0	0
trace	1	11	3	0	0
timedraw	1	11	3	0	0
endloop	0	0	0	0	0

==========

运　行　时　间

==========

本程序工作起始时刻: 2004年07月03日10时50分04秒

本程序工作终止时刻: 2004年07月03日10时50分17秒

五、节点位移文件HD000113.DAT

节点11的挠度随荷载变化如下。

0.10000000E+00　0.12412256E+01
0.20000000E+00　0.24006805E+01
0.30000000E+00　0.34284168E+01
0.40000000E+00　0.43044163E+01
0.50000000E+00　0.50348835E+01
0.60000000E+00　0.56385675E+01
0.70000000E+00　0.61371049E+01
0.80000000E+00　0.65504592E+01
0.90000000E+00　0.68954253E+01
0.10000000E+01　0.71855467E+01

与解析解相比，VFEAP给出的数值解偏大，主要是实体退化梁单元充分考虑梁的剪切变形。

参考文献

[1] 凌道盛.VFEAP使用手册.浙江大学建筑工程学院内部使用材料。

第13章

有限单元法在桥梁工程中的应用

桥梁结构往往跨径较大，结构复杂且超静定次数高，不可能单由精确的结构分析来完成。随着有限单元法的出现，将许多材料力学、结构力学及弹性力学中一系列困难问题几乎解决到完美的程度。桥梁工程与上述三大力学密切相关，因此，有限单元法对桥梁工程的发展作出了巨大的贡献。它几乎能解决桥梁工程设计施工中的一切问题，如结构静、动力空间结构分析、稳定分析、施工过程仿真和施工控制、桥梁结构优化设计、损伤识别、抗震分析、桥梁结构健康监测和状态评估、钢结构桥梁的疲劳评估、船桥碰撞响应及损伤分析等。这些问题可能是全桥的问题，也可能是个别构件甚至某一构件的其中一个节段的问题，都能成功地借助于计算机利用有限单元法来解决，并且其效率相当高。下面举几个用有限单元法对桥梁工程进行空间分析的实例，所有实例均采用 VFEAP 程序分析。

13.1 嘉王公路北郊河大桥的有限元分析[1] [2]

一、桥梁概况

嘉王公路北郊河特大桥位于嘉兴市北郊，是一座跨越京杭大运河及未来嘉兴市外环线的公路特大桥，全长 570.27 米，其中主桥跨径 51 米，为现浇全预应力混凝土简支箱梁桥，桥面宽度为 12.5（上行桥）+ 0.5（中央分隔带）+ 12.5（下行桥）m，其中预应力钢筋采用符合 ASTM 标准的高强度、低松弛钢绞线，与同类桥梁相比，该桥具有造型轻巧、经济技术指标高及施工工艺先进等特点，因而在桥梁工程界引起了广泛的关注。

51 米箱梁桥两端支撑采用盆式橡胶支座，可以简化成如图 13.1 所示的简支梁，沿桥长方向为分段截面，其中在 B 截面处腹板厚度 b 突变（见图 13.2）；而在 A 截面处箱梁底板厚度 a 变化，在 30cm 的长度范围内连续变化（见图 13.2，13.3）；其中箱梁高度为 2.00m，而桥面铺装层厚为 0.08m。箱梁承重结构的混凝土设计标号为 C50，桥面铺装混凝土设计标号为 C30。预应力钢筋采用按束分批两端张拉，每束由 5 根直径为 ϕ^j15.24 的钢绞线组成。主要的非预应力钢筋（通长布置的 66 根 ϕ16 钢筋）位于梁底板处，分上、下两排布置。荷载试验以交通 161 型 20 吨及交通 361 型 30 吨汽车模拟设计荷载，对 A、B、C 三个截面分别分级加载进行测试。本节按材料设计强度标号给出跨中（C 截面）加载时 B、C 测试截面的计算值，并与实测值及规范计算值进行比较。

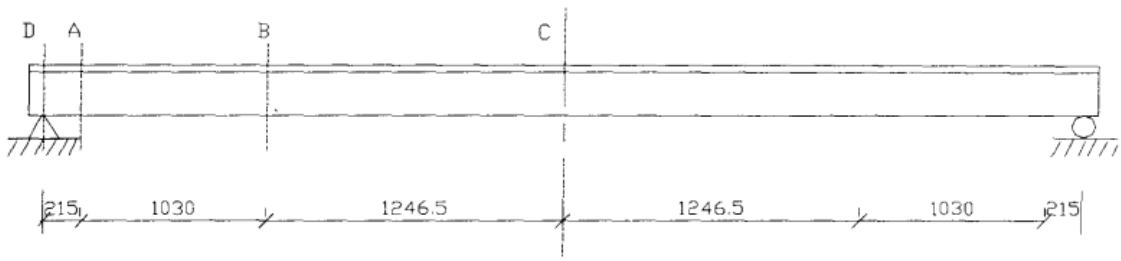


图 13.1　51m 箱型梁桥受力模型（尺寸单位：cm）

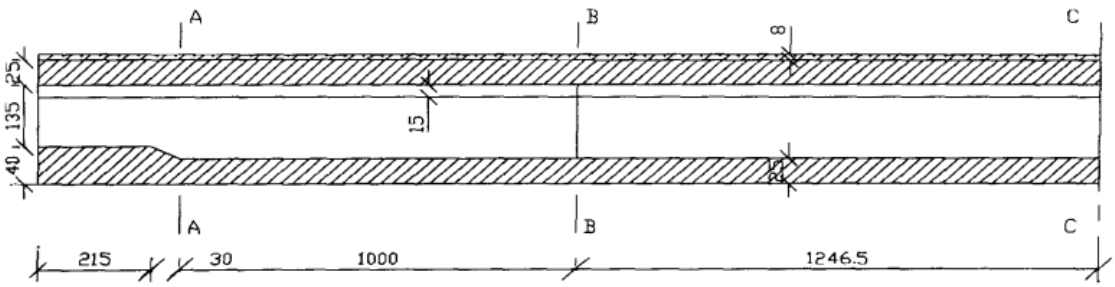
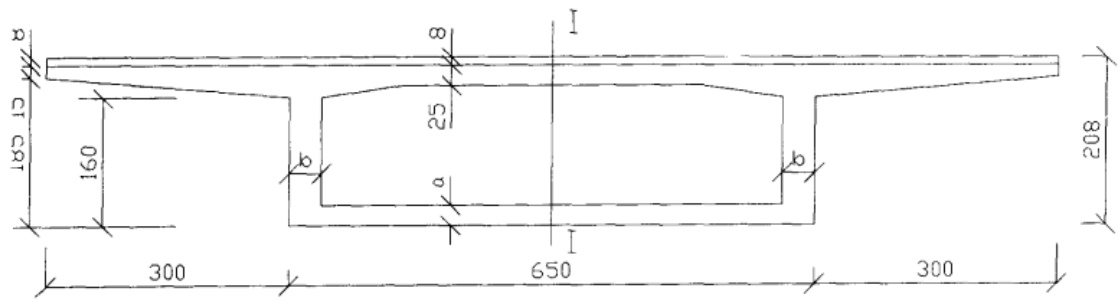


图 13.2　1/2 纵向剖面图（ I—I 剖面 ）（尺寸单位：cm）



注： *a*、*b* 为变化值，*a* 值在 *D*—*A* 段为 40，在 *A*—*C* 段为 25；*b* 值在 *D*—*B* 段为 45，在 *B*—*C* 段为 30。

图 13.3　箱梁各段横截面（单位：cm）

二、有限元计算模型

箱形梁是一种截面形式比较复杂的梁，梁的截面中有两种不同标号的混凝土，有作为预应力钢筋的钢绞线，有普通的二级钢筋。为计算方便，将底板普通钢筋（因为在不同的高度有两排，应分别考虑）、预应力钢筋及腹板预应力钢筋按面积折算为相应截面宽度的一薄层，将箱梁截面分层如图 13.4 所示。整个截面沿高度共分 14 层，自下而上依次是：（1）C50 混凝土，（2）普通钢筋，（3）C50 混凝土，（4）预应力钢筋，（5）C50 混凝土，（6）普通钢筋，（7）C50 混凝土，（8）C50 混凝土，（9）预应力钢筋，（10）C50 混凝土，（11）C50 混凝土，（12）C50 混凝土，（13）C50 混凝土，（14）C30 混凝土。其中（1）～（7）层的每层只有一种材料，具有同一宽度 650cm；（8）～（11）层的各层因为内部空缺，当作有两种材料，有不同宽度；（12）～（14）层的每层只有一种材料同一宽度，而（11）、（12）两层截面形状为梯形，其他层都为矩形截面。



图 13.4 箱梁 1/2 截面分层示意图

根据桥梁是分段变截面及桥梁测试截面位置的实际情况，将整个箱梁划分为 6×3×1 共计 18 个单元，采用箱梁两端底边简支的边界条件，如图 13.5 所示。考虑到箱梁上翼缘的底面是倾斜的，在构造虚拟层合单元真实材料与虚拟层的界面会是倾斜的，因此为保证在分层积分时同样有效，单元的虚拟节点抬高其坐标位置，如图 13.6 所示（实际上箱梁箱室内四角还有梗腋，见图 13.3，因尺寸相对较小，本文计算略去）。各虚拟层合单元均采用 16 节点单元（无板厚方向的边中节点），因单元内部分层不同，共有 4 种单元类型。其中桥梁两侧翼缘部分为一种单元，共分三层，自下而上依次是：虚拟层、梁体混凝土层、桥面铺装混凝土层；中间箱室部分沿桥梁长度方向分为三种类型单元，单元内部有梁体混凝土层、普通钢筋等效层、预应力钢筋等效层、箱室壁等效层、桥面铺装混凝土层等，具体分层参照图 13.6。

由于计算采用的单元尺寸较大，为保证计算所用的模拟荷载能够较精确地反映实际荷载，根据 8 节点等参元的形函数，对非节点集中力按二次插值方式计算相应单元的等效节点荷载，从而对相应节点加载。

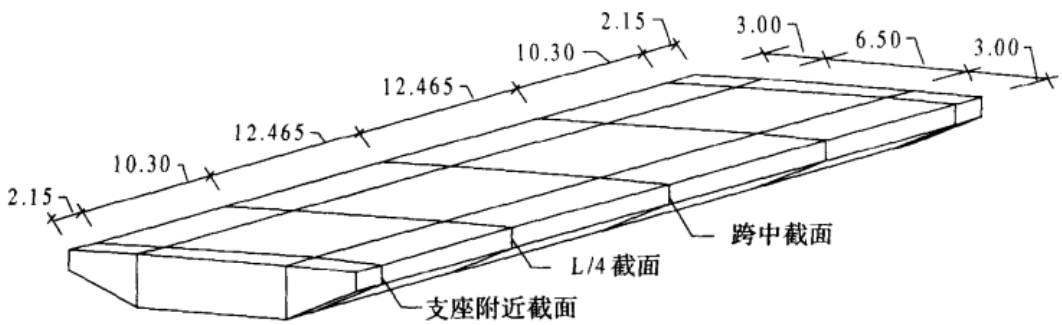


图 13.5 单元网格划分（尺寸单位：m）

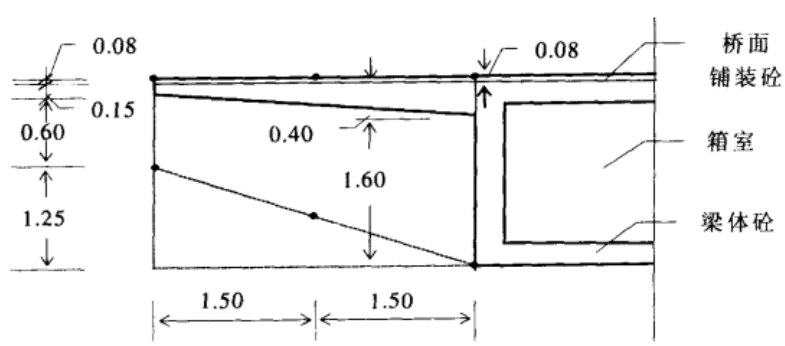


图 13.6 截面节点定义示意图（尺寸单位：m）

三、计算结果及其与试验值的比较

对称加载时，其挠度的计算结果列于表 13.1，混凝土及钢筋的应力列于表 13.2，作为比较，表中也列出了试验值及按规范的计算值。偏载时，箱梁的箱室左、右两侧壁下的测点处的挠度值也列于表 13.1。

表 13.1　挠度的计算值与试验测量值及规范计算值的比较　单位：mm

荷载等级	跨中截面			L/4 截面		
	计算值	试验值	规范值	计算值	试验值	规范值
一	14.71	12.38	26.54	10.32	8.13	22.16
二	16.23	13.54	29.19	11.28	8.99	23.78
三	17.74	14.83	31.85	12.24	9.75	25.39
偏载(左侧)	17.28	10.19	——	11.73	8.97	——
偏载(右侧)	18.24	13.10	——	12.27	9.65	——

表 13.2　钢筋及混凝土应力的计算值与试验测量值及规范计算值的比较　单位：MPa

测试截面	荷载等级	$\sigma_{下}$			$\sigma_{钢}$			$\sigma_{上}$		
		计算值	试验值	规范值	计算值	试验值	规范值	计算值	试验值	规范值
跨中截面	一	-2.696	-2.555	-4.664	-15.1	-22.0	-25.71	1.006	0.767	1.738
	二	-2.893	-2.750	-5.003	-16.9	-23.6	-27.58	1.076	0.810	1.864
	三	-3.090	-3.325	-5.343	-18.1	-24.6	-29.46	1.146	0.920	1.991
L/4 截面	一	-1.635	-1.365	-2.314	-10.6	-8.8	-12.73	0.658	0.632	1.019
	二	-1.757	-1.610	-2.472	-11.2	-9.2	-13.60	0.702	0.682	1.089
	三	-1.885	-1.645	-2.630	-11.9	-10.6	-14.47	0.745	0.778	1.159

四、结果分析及结论

由表 13.1 可以看出，按本方法的挠度计算值都稍大于实桥测试值。因实际上桥梁施工中可能存在一些跟设计期望值并不一致的因素（如抽检到的混凝土平均强度接近 60MPa 等），以及桥面系结构的影响等因素，作为理论控制值，应该说已经有较高的精度。由表 13.2 可以看出，混凝土应力的计算值与实测值十分接近，而钢筋应力的计算值与实测值有较大差距，这主要是因为计算只能采用钢筋与混凝土的应变协调模型，而受当时测试条件的限制，实桥钢筋应变的测量只能采用在一定部位开凿混凝土以露出钢筋并能打磨贴片的方式，这可能会导致该位置的钢筋产生局部的弯曲效应，还会导致该位置的钢筋及混凝土产生一定的应力集中效应，局部的应力场发生改变，并在局部范围内钢筋与混凝土之间有一定的不协调性等，这都会使钢筋的应力相对增大，而超过计算值。

由此可见，组合单元法可以用很少的单元和节点分析复杂的工程问题，并可以获得相当满意的精度，是一种非常适合于工程分析的计算方法。

13.2　宁波甬江大桥的有限元分析[3]

宁波市甬江大桥是横跨甬江，连接宁波市（甬）江东和（姚）江北二区的重要桥梁，其

主桥为一斜拉桥，跨径为 105m+97m，门式塔架，塔梁固结体系，拉索为 ϕ5 平行钢丝，热挤聚乙烯护索套，冷铸镦头锚具，用现浇混凝土挂篮悬臂施工工艺。塔墩基础采用桩径 1.6m 的钻孔灌注桩。全桥设计荷载：汽—20，挂—100，人群荷载 3.5kN/m^2。桥面宽度：车行道 15m，两侧人行道各 3m，主桥车行道两侧各有 2.5m 的索面净空，累计主桥宽 26m。

一、计算模型的建立

采用实体退化单元对甬江大桥进行离散，主梁单元采用 16 节点组合单元，横截面单元划分见图 13.7（共分为 9 个单元）；纵向每 8m（塔根段主梁 4m）划分为一个单元；主塔采用三维 20 节点实体退化梁单元，斜拉索采用杆单元。全桥离散为 308 个单元，1872 个节点。坐标系原点在桥塔处桥面中心点，x 轴为桥梁的纵向，正方向指向江东区，z 轴正方向垂直向上。建立的三维空间有限元计算模型详见图 13.8。

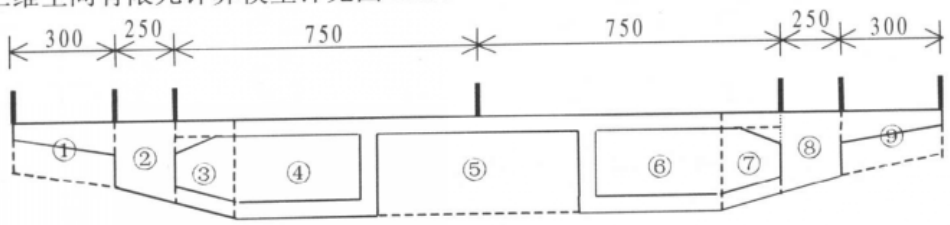


图 13.7 主梁横向单元划分示意图(单位：cm)

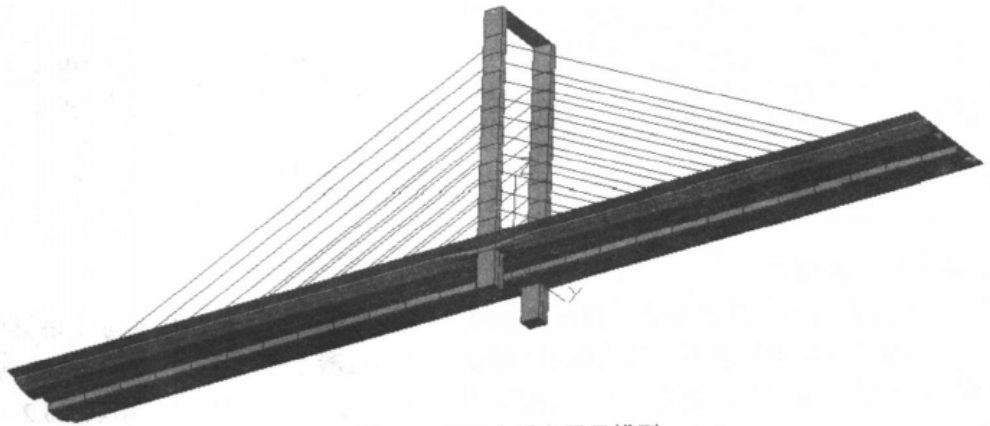


图 13.8 甬江大桥有限元模型

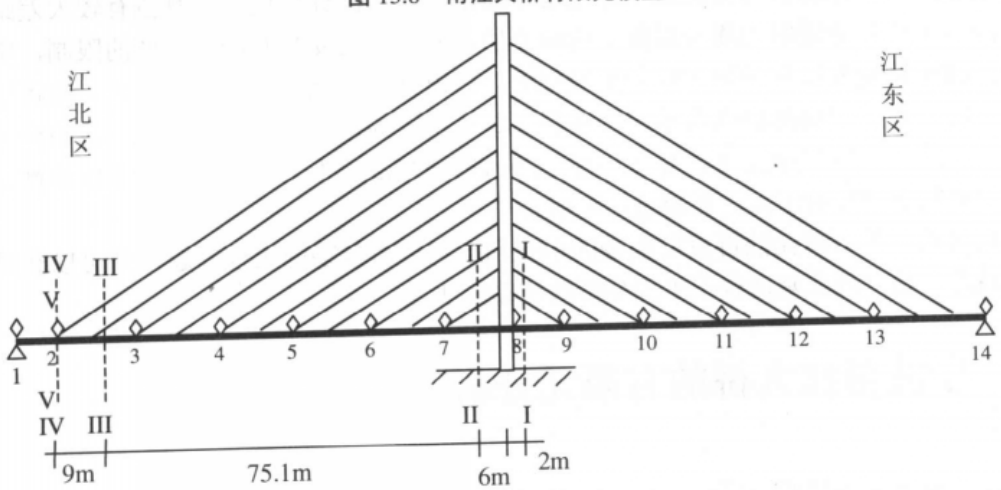


图 13.9 测试截面及挠度测点示意图

二、 测试截面、试验荷载及其分级

（1）测试截面

在建立的有限元计算模型上施加设计荷载，设计荷载分单侧、双侧、半桥、全桥、对称和偏载几种情况分别组合布载，找出设计荷载作用下各布载情况的最大应力值及其所在位置，并将该位置所在的截面作为测试截面，即图 13.9 中的Ⅰ、Ⅱ、Ⅲ、Ⅳ截面。

（2）试验荷载

试验采用汽车进行分级加载，加载车辆选用 12 辆 ST300 型载重汽车，前轴重 60kN，中、后轴各重 120kN。前轴至中轴的距离为 3.85m，中轴至后轴的距离为 1.30m。

（3）荷载分级

试验荷载分三级，分别约为设计控制内力的 80%、90%、100%的控制荷载。

（4）纵向加载位置的确定

纵向车辆的布载按加载截面最大应力点纵向影响线试算确定，各试验加载工况纵向车辆的布置详见图 13.10 和表 13.3。

（5）横向加载位置

Ⅰ截面的横桥向车辆布载采用半桥偏载形式，Ⅱ、Ⅲ、Ⅳ截面横桥向车辆布载采用半桥对称靠边布置形式。

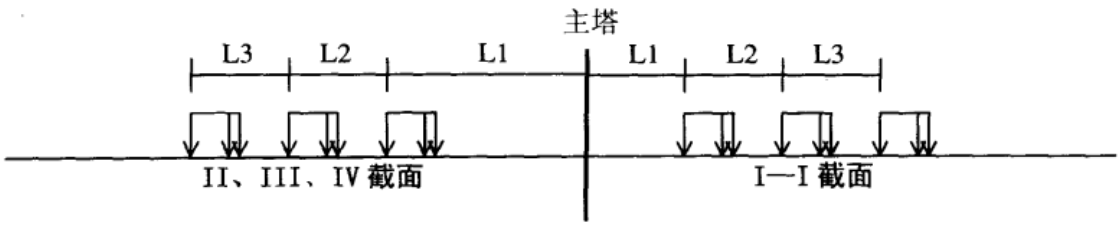


图 13.10　加载车辆位置示意图

表 13.3　各加载工况下 L1、L2、L3 值

加载截面	加载工况	L1	L2	L3	加载截面	加载工况	L1	L2	L3
I	80%加载	12.49	9.66		III	80%加载	81.85	14.3	
	90%加载	12.49	9.66	59.57		90%加载	54.1	27.8	14.3
	100%加载	12.49	9.66	45.07		100%加载	64.2	17.7	14.3
II	80%加载	17.35	16.5	7.7	IV	80%加载	89.9	11.3	
	90%加载	25.09	8.76	7.7		90%加载	57.1	32.8	11.3
						100%加载	69.2	20.7	11.3

注：II 截面由于受加载车辆数量的限制，只能加载 80%、90%二种工况。

三、测点布置

(1) 应变测点

采用电阻应变片（胶基）为应变传感器，应变采用上海华东电子仪器厂生产的 YJ—22 静态电阻应变仪检测；并在关键点辅以千分表作双重保护测量，以保证测试数据的可靠性。

(2) 挠度测点

采用连通管测量，测点布置在上游侧的防撞护拦上，测点的纵向位置示于图 13.11（共 14 个测点）。

(3) 塔顶偏移

塔顶在顺桥向的偏移采用激光经纬仪检测。

(4) 测试截面测点

各测试截面测点的布置见图 13.11。对 I、II 截面各布置了 14 个应变片；对 III、IV 截面，由于以受弯为主，取消了图 13.11 中的 9～14 号应变片。

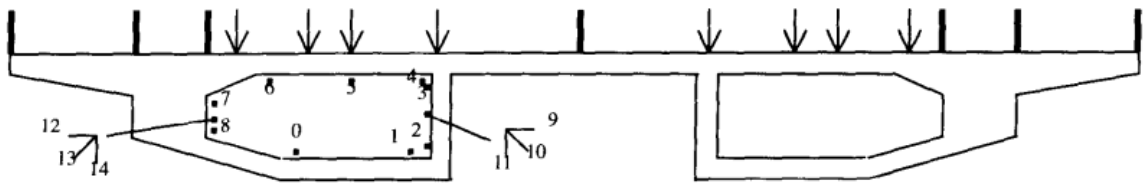


图 13.11 各截面测点布置

四、有限元分析结果及与荷载试验结果的比较

(1) 挠度、应变影响线

桥梁结构的实测影响线反映了桥梁结构中某一测试点的应力、应变或变形在（单位）移动荷载作用下的变化情况。如果正常运行状态下的各种效应值在规范规定的范围内，则可利用该影响线按迭加原理进行实际承载能力的推定。

本次试验采用单排四辆 ST300 型载重汽车作为移动荷载。横桥向采用图 13.11 所示的对称靠边布载，纵桥向的布载为江东区跨每隔 8m 布载一次，江北区跨每隔 4m 布载一次。各次布载时均检测各控制测点的效应值。

这里，挠度影响线只选取了有代表性的 3#（－74m）、12#（42m）两个测点的挠度影响线（图 13.12～13.13）；应变影响线只列出了 I～IV 截面顶板或底板中一个测点的影响线，见图 13.14～13.17。各图中均同时列出有限元计算曲线，以资比较。

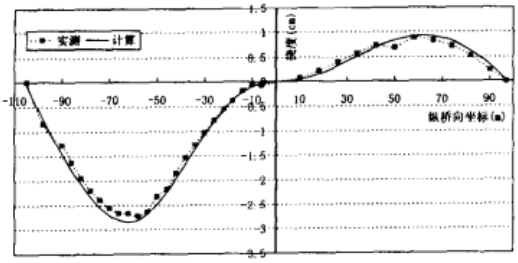


图 13.12 3#测点(–74m)挠度影响线

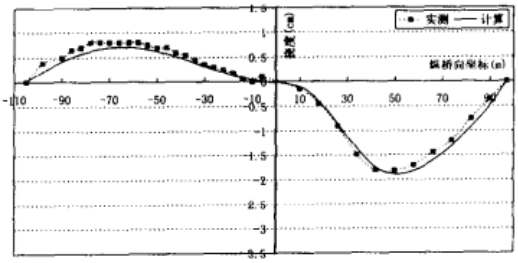


图 13.13 12#测点(42m)挠度影响线

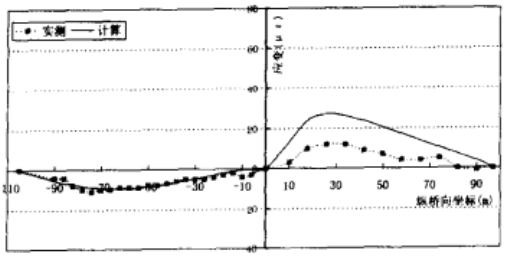


图 13.14 I 截面 5#点应变影响线

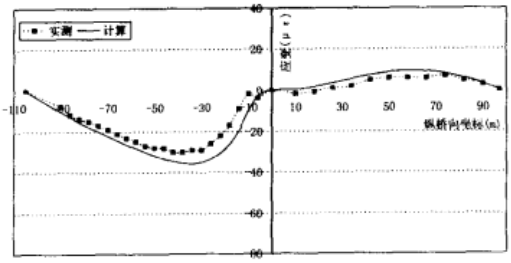


图 13.15 II 截面 1#点应变影响线

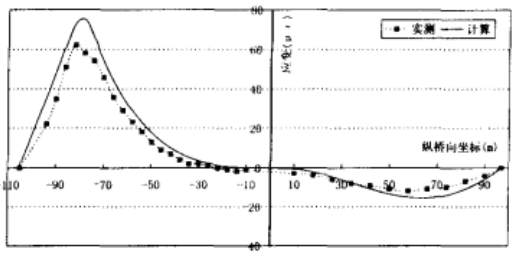


图 13.16　III 截面 0#点应变影响线

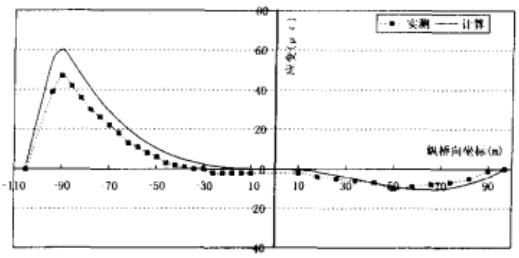


图 13.17　IV 截面 1#点应变影响线

从挠度和应变影响线图可以看出：

第一、实测影响线与计算影响线无论是形状还是峰值的位置均较吻合，说明有限元计算模型基本符合结构的实际状态；

第二、实测挠度影响线与计算影响线很接近，这说明有限元模型能较好地反映桥梁结构变形情况；

第三、实测应变均小于计算值，说明桥梁的实际刚度大于理论计算模型。

（2）挠度

主梁各截面在各级荷载作用下的主梁挠度曲线见图 13.18～13.21。

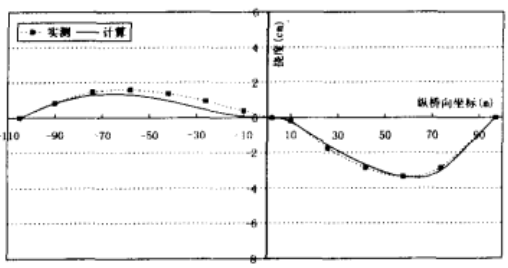


图 13.18　I－I 截面 100%加载挠度曲线

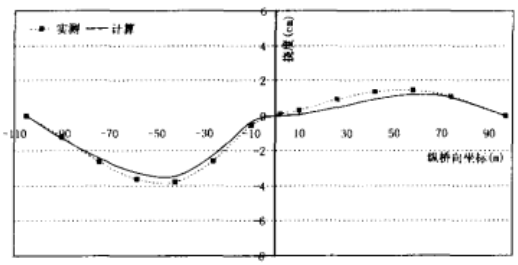


图 13.19　II－II 截面 100%加载挠度曲线

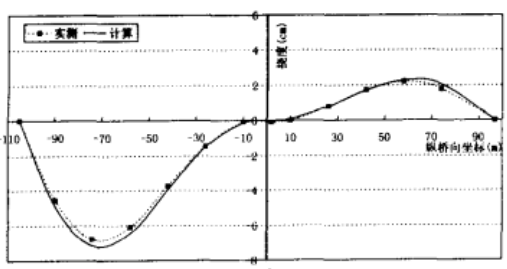


图 13.20　III－III 截面 100%加载挠度曲线

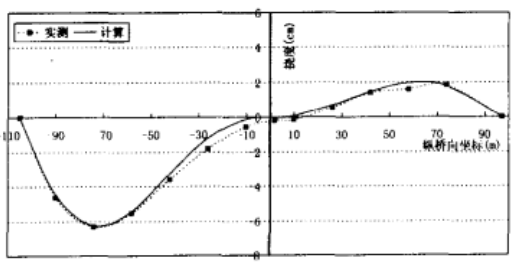


图 13.21　IV－IV 截面 100%加载挠度曲线

(3) 塔顶偏移

在 III—III 截面 100%加载时，塔顶偏移值为最大，实测 27mm，理论计算值为 29mm，相差 6.9%。

(4) 固有频率

利用上述模型数值分析了桥梁前 25 阶固有频率，同时利用环境激励测出部分固有频率，其中前 20 阶的计算结果和测试结果比较见表 13.4。

表 13.4 甬江大桥计算频率和实测频率比较（单位：Hz）

频率阶次	1	2	3	4	5	6	7	8	9	10
计算结果	0.192	0.665	1.186	1.396	1.420	1.519	1.568	2.022	2.104	2.353
实测结果		0.600	0.975				1.613	1.988	2.100	
相对误差		10.9%	21.7%				-2.79%	1.72%	0.175%	
频率阶次	11	12	13	14	15	16	17	18	19	20
计算结果	2.379	3.523	3.737	3.847	4.035	4.150	4.230	4.457	4.478	4.514
实测结果		3.488				4.125				
相对误差		1.01%				0.605%				

由于低频测试的困难，第二、第三阶频率相对误差较大，其他的计算频率和实测频率误差均较小。经过部分实测结果验证为正确的有限元模型，可以计算出各阶频率和模态，可以进行动力响应分析。正确的建模也是施工控制的关键。用三维实体退化虚拟层合单元，使得像箱梁桥这样复杂的空间结构的静、动力分析和施工控制变得简单、有效。

13.3 香港青马大桥的振动分析[4]

青马大桥是一座公路、铁路两用的悬索桥，大桥全长 2160m，主跨长度 1377m，于 1997 年完成通车。其总体布置如图 13.22 所示。

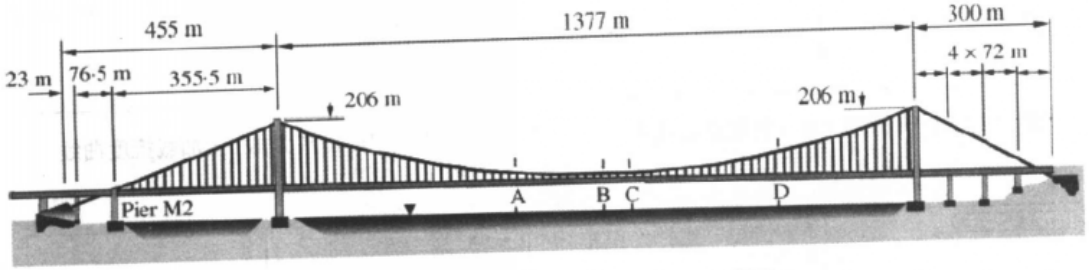


图 13.22 青马大桥总体布置示意图

一、有限元分析模型的建立

本文主要分析青马大桥的动力特性，因此，以成桥状态作为建模对象。整体坐标系原点取在马湾塔中心线标高±0.00m 处。马湾指向青衣方向（纵桥向）为 x 轴正方向。并设此方向为前进方向，前进方向左侧为 y 轴正方向，垂直向上为 z 轴正方向。规定了整体坐标系后，就可以将桥上任一点坐标计算出来。

（1）主塔

主塔为钢筋混凝土多层门式框架结构，塔柱为变壁厚圆端形单箱双室结构（如图 12.23），塔高 206m。每个塔柱宽 6m（沿横桥向），纵桥向尺寸从 18m 缩小到 9m。塔柱竖向倾斜坡度为 1 / 100，并用预应力混凝土门式深梁连接。详见图 13.24。

为了减少总体计算工作量，要求剖分单元数量尽量少，同时，为了充分发挥组合单元的优势，在横截面方向，每根塔柱只用一个组合单元，如图 13.23 中的 1、4、12、9，然后在组

合单元内分成五个区域，分别为两个圆端形 2、6、10、5 和 3、8、11、7；矩形 2、3、11、10，这三个区域均采用 20 节点三维实体。另外两个区域分别为矩形 13、14、18、17 和 15、16、20、19，这两块区域材料是空的，但由于前面三块区域中已经包含了这两块区域，因此应该扣去，只要在输入材料特性时，将这两块区域的某些材料特性参数（如弹性模量、剪切模量等）设为负值即可，由于这两块区域也是变截面，因此也采用 20 节点三维实体。

在塔柱高度方向，根据塔柱横截面变化规律，并考虑与横梁的连接，共划分了 15 个组合单元。

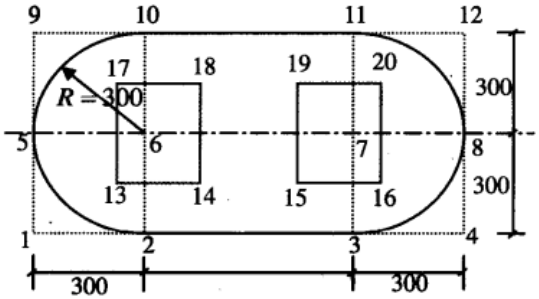


图 13.23　塔柱单元划分示意图

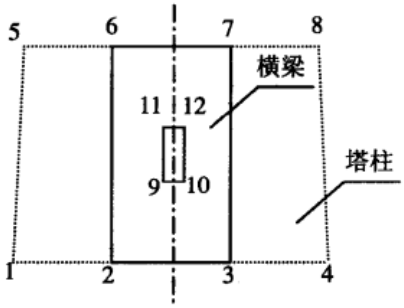


图 13.24　横梁单元划分示意图

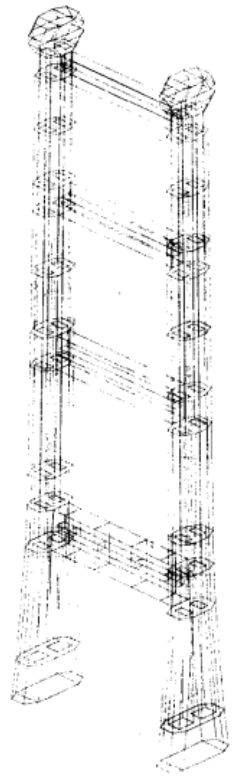


图 13.25　主塔建模效果图

横梁截面虽然为矩形，但其宽度比塔柱要小很多。由于塔柱在横截面方向只剖分了一个单元，因此，应将组合单元 1、4、8、5 的宽度扩大到与塔柱同宽，以便于与塔柱单元之间的连接，如图 13.24 所示。组合单元内部划分为两块区域，正材料区域 2、3、7、6 和负材料区域 9、10、12、11。沿横梁长度方向，除考虑到下横梁与加劲梁的连接而划分为五个单元外，其余横梁均只划分为一个单元。因此，一个主塔划分为 38 个单元。建模效果如图 13.25 所示。

（2）缆索

缆索主要包括主缆和吊索，采用大位移空间杆单元模拟。青马大桥主跨和 359m 边跨有吊索，96m 和 300m 边跨没有吊索。对于吊索，一根吊索取为一个杆单元，全桥共 190 个单元。对于有吊索的主缆，取两根吊索间主缆取为一个单元，即水平向长度为 18m (主塔附近两根为 13.5m)。对于没有吊索的主缆，可均分为与有吊索主缆相近的长度。

关于主缆的线形，对于有吊索段，假定其线形为抛物线；对于无吊索段，假定其线形为悬链线。

青马大桥成桥状态主缆线形及各控制点标高详见图 13.26。

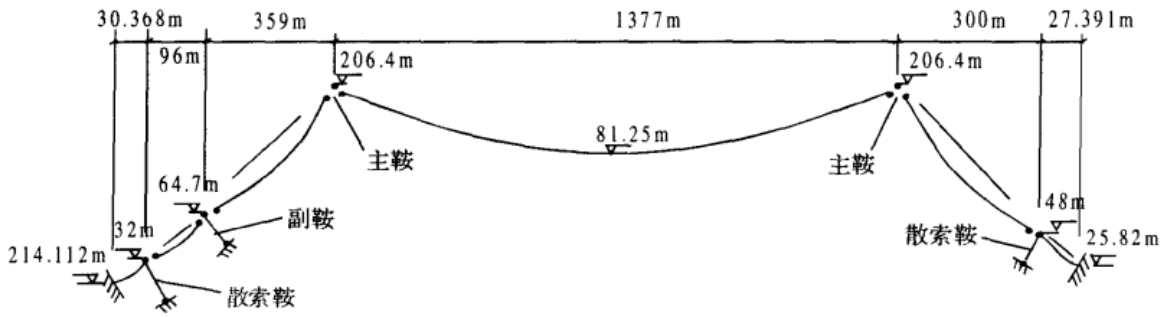


图13.26 成桥状态主缆线形示意图

（3）加劲梁

根据组合单元划分单元的原则，在横截面方向，将加劲梁分为七段，同时，为了便于计算内部各构件节点的局部坐标，组合单元一律取为矩形（如图 13.27 所示），分别为左侧吊索段 1、2、10、9，左侧外段 2、3、11、10，左侧内段 3、4、12、11，中段 4、5、13、12，右侧内段 5、6、14、13，右侧外段 6、7、15、14，右侧吊索段 7、8、16、15。

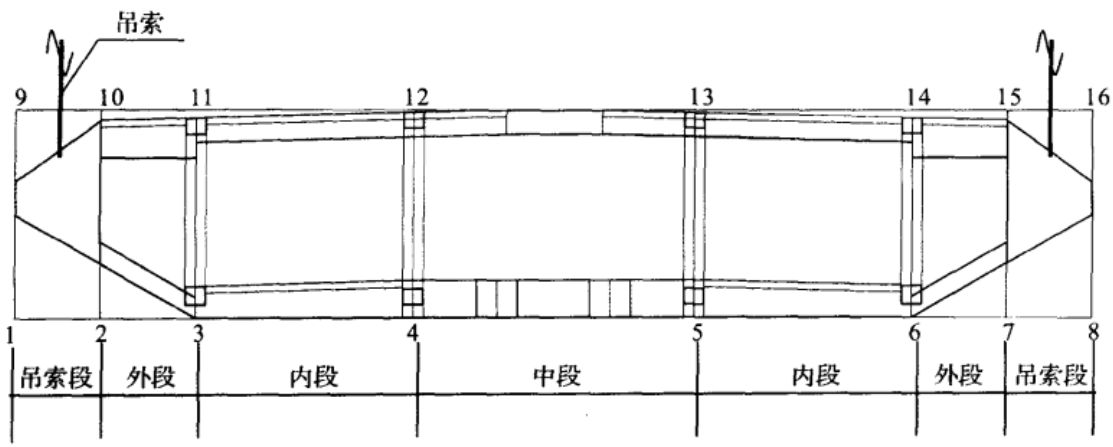


图 13.27 加劲梁横截面方向单元划分示意图

在纵桥向，加劲梁的整体受力特性应表现为空间梁的特性，只要划分适当，用 20 节点空间等参单元足以描述，除此以外，还应考虑加劲梁与其他单元之间的连接和边界条件。在主跨和有吊索的 355.5m 边跨，取两根吊索之间为一个单元，其余跨一般取 18m 为一个单元。详细分段见图 13.28，整个加劲梁在纵桥向共划分为 122 段，每段在横桥向划分为 7 个单元，因此共划分了 122×7＝854 个单元。

桁架式加劲梁内部主要由板条组成，对各板条按其所在空间位置和材料特性分别计算其对单元刚度矩阵的贡献。对于桥面板下的 U 形加劲肋，本文采用一层等效板代替，如图 13.29 所示。加劲梁建模效果图详见图 13.30。

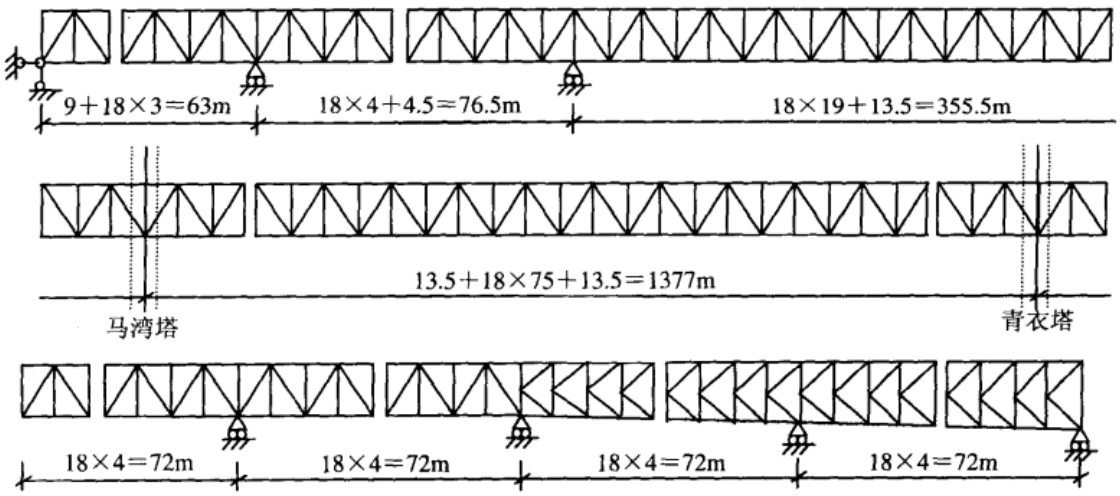


图 13.28　加劲梁在纵桥向单元划分示意图

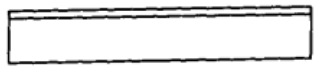


图 13.29a　U 形加劲肋

图 13.29b　等效 U 形加劲肋

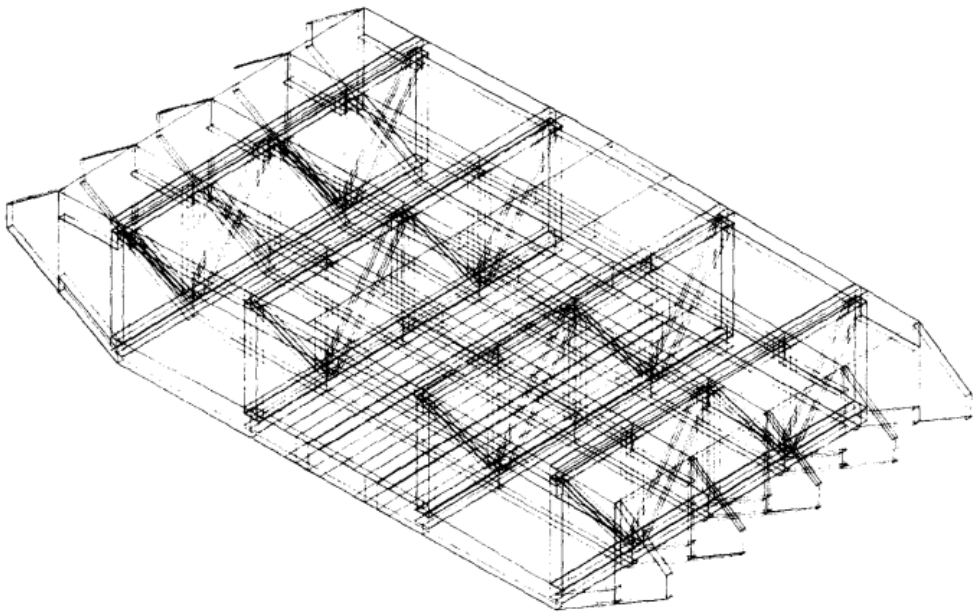


图 13.30 加劲梁段建模效果图

（4）主鞍、副鞍和散索鞍

成桥状态时，主缆在主鞍上是被固定不动的，主鞍与桥塔之间也无相对位移，主缆在纵桥向的位移通过主塔的柔性（塔顶在纵桥向的位移）实现。因此，主鞍可作为塔的一部分来考虑，只是其力学性质参数与塔的不同。

成桥状态时，主缆在副鞍、散索鞍上也是被固定不动的，但副鞍、散索鞍与主鞍不同，其下部没有很高的桥塔。为了实现主缆在张力变化时产生的位移，散索鞍下被设计成摆柱形式。由于主缆采用了大位移杆单元，为了与主缆连接的方便，本章用一个轴向刚度很大的杆单元来模拟散索鞍铰接摆柱作用，详见图 13.26，这样模拟还可以分析副鞍、散索鞍底座受力情况及其转角位移情况。

（5）边界条件的确定及各单元间的连接

边界条件对悬索桥的动力特性会产生较大的影响。因此，在建立悬索桥有限元计算模型时应正确模拟边界连接条件。对于青马大桥，边界条件的处理如下：

1、悬索桥的大缆在两端通过非常强大的锚碇固定，因此，大缆两端作固定处理。大缆与主塔作固结处理，与副鞍、散索鞍铰接，副鞍、散索鞍的另一端与大地间作铰接处理。

2、两主塔的基础分别为深基坑和沉箱，顶部又有强大的承台与塔柱连接。因此，每根塔柱底部各节点作固定处理。同样地，各墩台在承台处均作固定处理。

3、加劲梁在马湾台、M1 墩、M2 墩、主塔、T3、T2、T1、青衣台处支承见表 13.5 及图 13.31。所有的支承均用杆单元模拟，杆单元一端连接加劲梁，另一端与相应的墩、台、主塔连接。

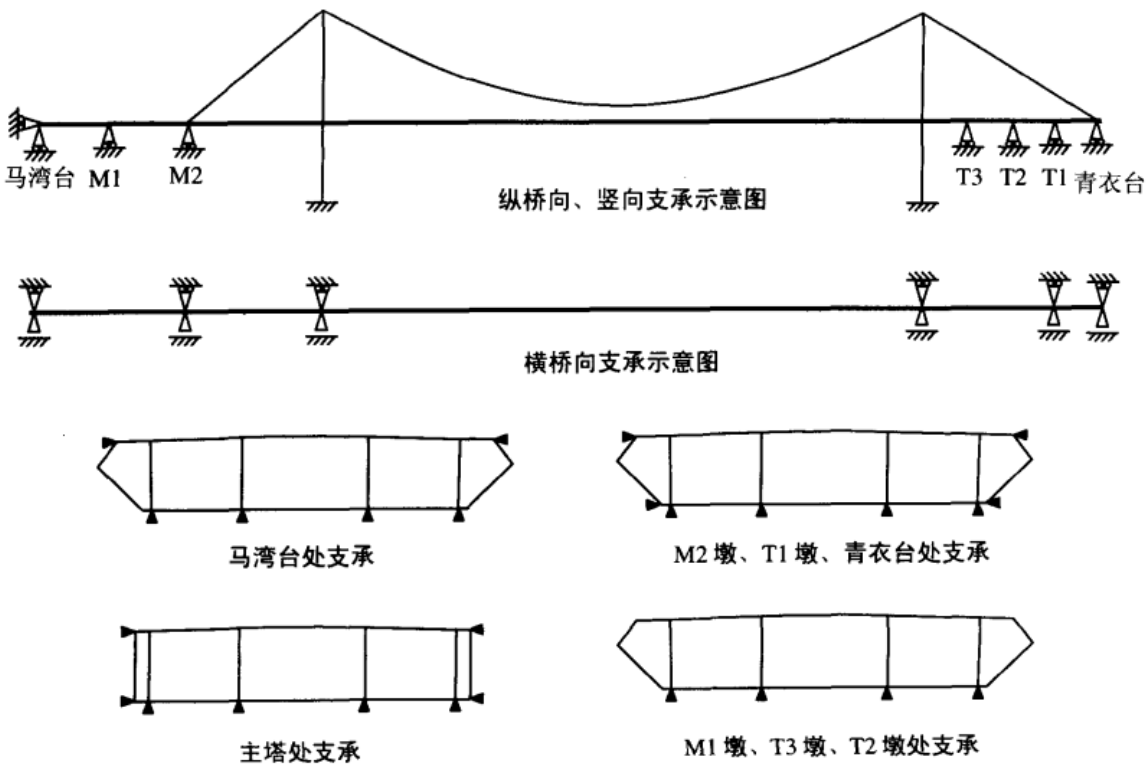


图 13.31 加劲梁支承示意图

表 13.5　加劲梁支承一览表

墩台名称	竖向支承	横桥向支承	纵桥向支承
马湾台	√	√	√
M1 墩	√		
M2 墩	√	√	
马湾塔	√	√	
青衣塔	√	√	
T3 墩	√		
T2 墩	√		
T1 墩	√	√	
青衣台	√	√	

（6）全桥有限元计算模型

按照上述处理，建立青马大桥有限元振动计算分析模型如图 13.32。全桥各主要构件的划分单元数、节点数和自由度数见表 13.6，为了便于比较，表中还列出了文献[5]用 ABAQUS 程序的杆系有限元建模的情况，由于文献[5]对大部分构件均采用了二节点空间梁单元，因此，虽然节点数较少，但自由度数很多，如果对总自由度数进行比较，则本文总自由度数只有文献[5]的 58%。体现了组合单元的优越性。

表 13.6　青马大桥有限元模型单元数、节点数和自由度数

构件名称	主塔	主缆	吊索	加劲梁	桥台、墩	连接杆	合计	ABAQUS
单元数	76	260	190	854	72	58	1510	19169
节点数	880	258		6626	852		8616	7699
自由度数	2640	774		19878	2556		25848	44562

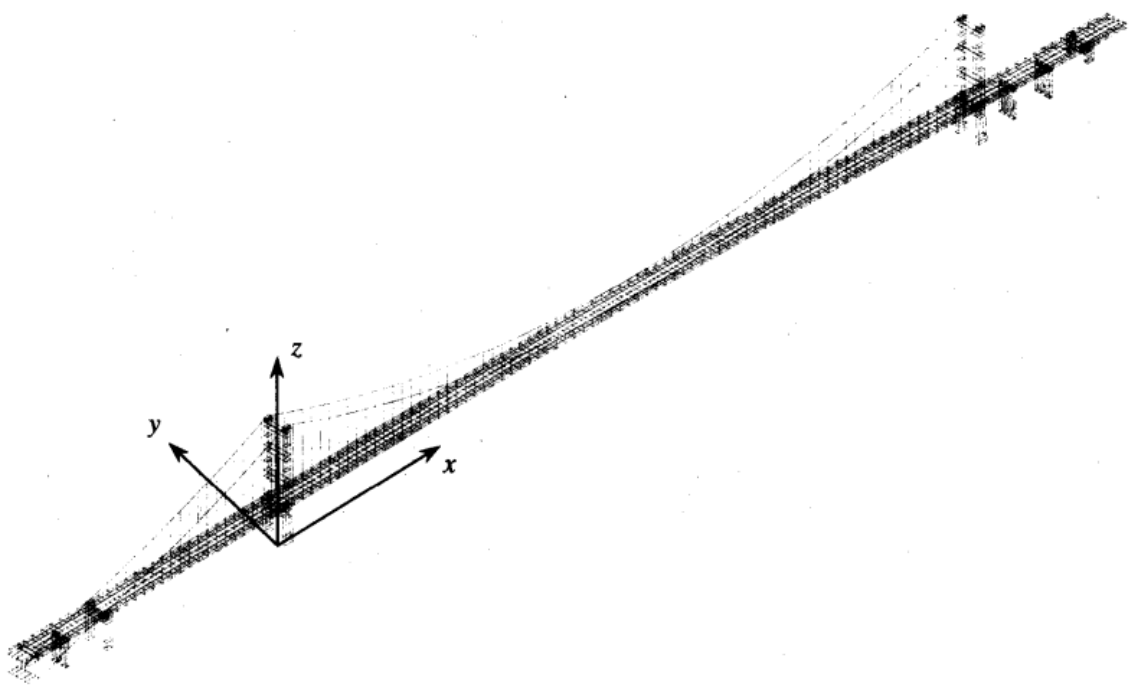


图 13.32　青马大桥有限元计算模型

二、成桥状态缆索初应力的计算

考虑到本例的主要目的是计算成桥状态下青马大桥的振动特性，只需要计算成桥状态下的主缆及吊索的初应力。因此，从设计成桥状态出发，提出假设—计算—比较—修改假设的多次迭代法。即先假设主跨和马湾边跨有吊索段的大缆线形为抛物线，其余边跨无吊索大缆的线形为悬链线，并假设大缆初应力为一恒定初值，计算恒载作用下的大缆和吊索应力，以此计算值作为下一步计算的初应力，如此循环，不断修改主缆和吊索的初应力，直至计算的大缆线形控制点的坐标与设计成桥状态基本吻合为止。在计算中，考虑到二期恒载所占比重较小，加劲梁段之间不刚接，即假设加劲梁各单元材料的弹性模量为小值。然后，以末次计算的主缆和吊索应力作为初应力，并将加劲梁刚接，即恢复加劲梁各单元的弹性模量至正常值，再进行振动计算。

三、青马大桥振动特性分析

（1）计算频率及振型

根据获得的成桥状态，采用组合单元刚度矩阵和一致质量阵，将动力学方程转化为式(13.1)的广义特征值问题，用逆迭代法求解，并在计算中采用“移频”[6]技术，求得了青马大桥前44阶自振频率和振型，详见表13.7。

$$\left(\boldsymbol{K}-\omega^2\boldsymbol{M}\right)\boldsymbol{d}=\boldsymbol{0} \tag{13.1}$$

式中，$\boldsymbol{K}$、$\boldsymbol{M}$为成桥状态结构的刚度矩阵和一致质量阵。

为了便于比较，表13.7中还列出了文献[5]的计算结果，从表中可知，除前几阶计算振动频率及振型一致外，两者很难作直接的比较，其中明显的差异在于，本文计算结果中有桥墩的振型，而文献[5]则没有，说明两者所建模型存在一定的差异。但是从振型上看，两者的计算结果存在可比性，因此，可进行分类比较。

表13.7 计算结果及比较

阶次	文献[5]		本文	
	频率(Hz)	振型特点	频率(Hz)	振型特点
1	0.06826	主跨侧弯0.5波，主缆面外	0.07080	主跨侧弯、主缆面外0.5波
2	0.11933	主跨竖弯1波，主缆面内	0.11937	主跨竖弯1波、主缆面内
3	0.14389	主跨竖弯1.5波，主缆面内	0.14013	主跨竖弯1.5波、主缆面内
4	0.15849	主跨侧弯1波，扭转1波	0.17025	主跨侧弯、扭转1波、主缆面外
5	0.19479	主缆局部异相摆振0.5波	0.18992	主跨、马湾边跨竖弯2波、主缆面内
6	0.20032	主跨、马湾边跨竖弯，主缆面内	0.23163	主缆面外异相摆振0.5波
7	0.21765	主缆异相摆振1波	0.23821	主缆面外异相摆振1波
8	0.21907	主缆同相0.5波，主跨侧弯	0.24958	马湾、主跨大缆面外同相摆振0.5波、梁侧弯并扭转
9	0.22096	主缆同相1波，主跨扭转并侧弯	0.25151	主缆面外同相1波、主跨侧弯并扭转
10	0.24242	主跨侧弯并扭转，主缆面内面外	0.25329	主跨、马湾边跨竖弯振动、主缆面内
11	0.24775	主跨扭转，主缆面内	0.27464	马湾边缆面外同相0.5波
12	0.25255	主跨、马湾边跨竖弯2.5波，主缆面内	0.27920	马湾边缆面外异相0.5波
13	0.27121	主跨侧弯并扭转，主缆面外	0.28855	T3墩纵桥向弯曲振动
14	0.29813	主跨、马湾边跨竖弯3波，主缆面内	0.30106	主跨、马湾跨竖弯3波、主缆面内

15	0.32280	主缆局部异相摆振 1.5 波	0.30935	T2 墩纵桥向弯曲振动
16	0.32971	马湾边缆局部同相摆振 0.5 波	0.31040	主跨、马湾边跨侧弯并扭转 2 波、主缆面外
17	0.33004	主跨、马湾边跨竖弯 3.5 波，主缆面内	0.34070	主跨扭转 0.5 波、主缆面外
18	0.33677	主跨扭转并侧弯，马湾边缆局部同相摆振	0.35150	青衣边缆面外同相 0.5 波
19	0.34163	马湾边缆局部异相摆振 0.5 波	0.35177	主跨、马湾边跨竖弯 3.5 波、主缆面内
20	0.37242	主缆局部同相摆振 2 波，青衣塔摆振	0.35413	M1 纵桥向弯曲振动
21	0.38277	主跨侧并扭转	0.36121	青衣边缆面外异相 0.5
22	0.38565	青衣边缆局部异、同相摆振 0.5 波	0.38604	主缆面外异相摆振 1.5 波
23	0.38798	主缆局部异相摆振 2 波	0.40321	青衣边缆面内同相 0.5 波
24	0.39434	青衣边缆面外异相摆振 0.5 波	0.40395	青衣边缆面内异相、主缆面外同相摆振 2 波、主跨侧弯并扭转
25	0.39825	青衣边缆面内同相摆振 0.5 波	0.40528	青衣边缆、主缆面外同相摆振 2.5 波。主跨侧弯并扭转
26	0.39903	青衣边缆面内异相摆振 0.5 波	0.41714	主缆面外异相摆振 2 波
27	0.40912	主跨侧弯并扭转，主缆面内面外	0.41751	主缆面外同相摆振 2 波。全跨侧弯并扭转
28	0.42394	主跨+马湾边跨竖弯 4 波，主缆面内	0.44162	主跨、马湾边跨竖弯 4 波，主缆面内
29	0.45106	马湾边跨扭转+侧弯	0.44174	主跨、马湾边跨侧弯并扭转 2 波，主缆面外同相
30	0.46351	主缆局部同相摆振 2.5 波，主塔同相摆振	0.45079	加劲梁纵漂、竖弯
31	0.47871	主缆局部异相摆振 2.5 波	0.46859	马湾塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转
32	0.50118	主塔异相摆振	0.47501	M2 墩纵桥向振动
33	0.50254	主跨+马湾边跨竖弯 4.5 波，主缆面内	0.49321	青衣塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转
34	0.51958	主跨扭转+侧弯，主缆面内面外	0.50717	马湾边缆面外异相摆振 1 波
35	0.52672	主塔同相摆振，主缆+边缆同相摆振	0.51010	马湾边跨、主跨侧弯 2 波，M2 墩横桥向振动，主缆面外同相
36	0.55672	主缆局部同相摆振 3 波	0.52043	主跨扭转 1 波
37	0.56162	主缆局部异相摆振 3 波	0.52583	马湾跨、主跨侧弯并扭转，马湾塔横桥向摆振，主缆面外同相
38	0.58439	主塔局部异相摆振	0.53020	主缆面外异相摆振 2.5 波
39	0.61948	主跨+马湾边跨竖弯 5 波，主缆面内	0.55168	马湾跨、主跨竖弯 4.5 波，主缆面内
40	0.64759	主跨扭转+侧弯	0.55212	主跨侧弯并扭转 2.5 波，青衣塔横桥向振动，主缆面外同相
41	0.64946	主塔同相弯曲振动（马湾塔为主）	0.56426	主塔纵桥向同相弯曲振动

42	0.65095	主缆局部异相摆振 3.5 波	0.56775	主跨、马湾跨侧弯并扭转 3 波，主缆面外同相
43	0.65437	主缆局部同相摆振 3.5 波	0.58973	主塔纵桥向异相弯曲振动
44	0.66793	马湾边跨竖弯，主缆面内	0.59364	主缆面外异相摆振 3 波

(2) 计算振型的分类比较

青马大桥的计算振型可分为以下六种主要振型：

① 加劲梁竖向弯曲振动

加劲梁竖向弯曲振动计算结果比较见表 13.8，在大部分振型中，马湾边跨也参与了主跨的竖向振动，但青衣边跨则没有，这是因为马湾边跨有吊索，而青衣边跨的加劲梁由桥墩支承，虽然加劲梁是连续的，但其变形又受到了桥墩支承的约束。在加劲梁竖向弯曲振动的同时，主缆一般表现为面内同相振动。图 13.33～13.36 列出了有代表性的第 2、3、5、10 阶振型。

表 13.8 加劲梁竖向弯曲振动计算结果比较

文献[5]			本文		
阶次	频率(Hz)	振型特点	阶次	频率(Hz)	振型特点
2	0.11933	1 波	2	0.11937	1 波
3	0.14389	1.5 波	3	0.14013	1.5 波
6	0.20032	2 波，马湾跨参与	5	0.18992	主跨、马湾边跨 2 波
12	0.25255	2.5 波，马湾跨参与	10	0.25329	主跨、马湾边跨 2.5 波
14	0.29813	3 波，马湾跨为主	14	0.30106	主跨、马湾跨 3 波
17	0.33004	3.5 波，马湾跨参与	19	0.35177	主跨、马湾边跨 3.5 波
28	0.42394	4 波，马湾跨参与	28	0.44162	主跨、马湾边跨 4 波
33	0.50254	4.5 波，马湾跨参与	39	0.55168	主跨、马湾边跨 4.5 波

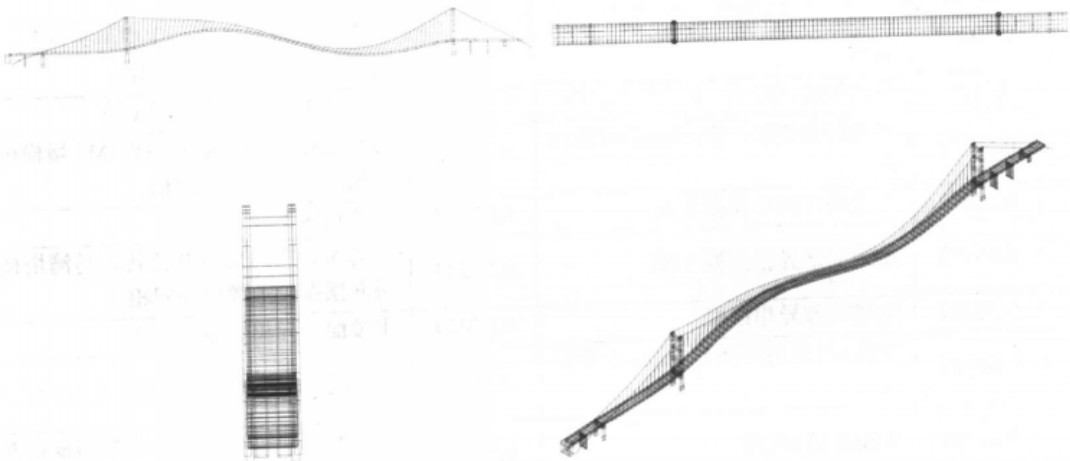


图 13.33 第 2 阶：主跨竖弯 1 波、主缆面内（f =0.11937Hz）

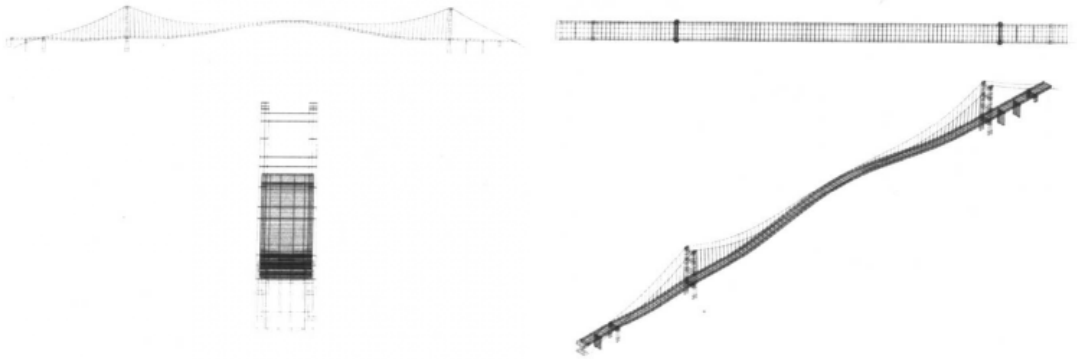


图 13.34　第 3 阶：主跨竖弯 1.5 波、主缆面内（f =0.14013Hz）

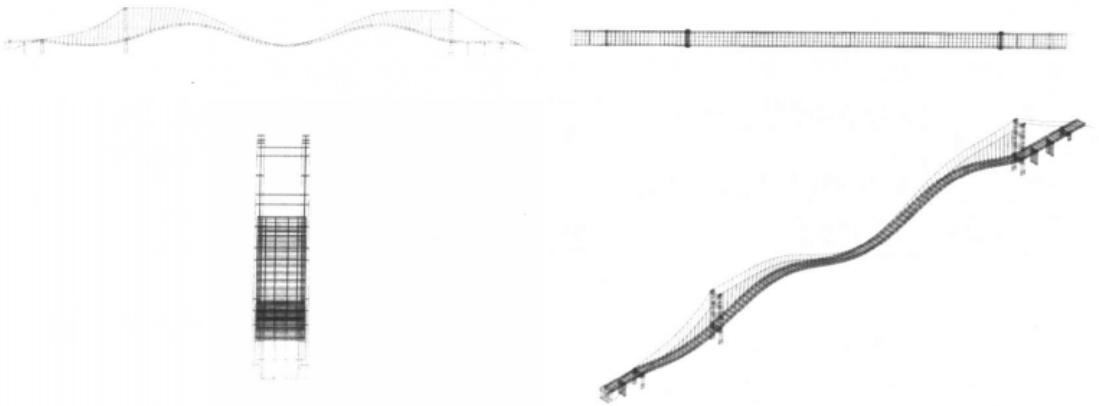


图 13.35　第 5 阶：主跨、马湾边跨竖弯 2 波、主缆面内（f =0.18992Hz）

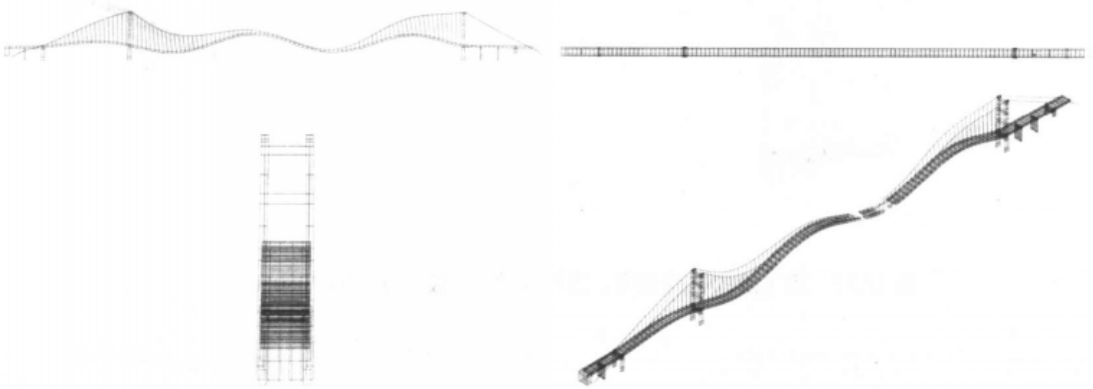


图 13.36　第 10 阶：主跨、马湾边跨竖弯 2.5 波，主缆面内（f =0.25329Hz）

②加劲梁侧向弯曲和扭转振动

由于侧向弯曲往往与扭转振动耦合，因此，把它们放在一起进行比较（表 13.9）。当主缆与加劲梁侧弯的方向一致，并且侧弯的幅度一样时，才会出现纯侧弯的振型，如第 1 阶，这一阶振型的频率是青马大桥所有振型中最低的，即第一阶振型。文献[5]计算结果中只有一阶振型（第 11 阶）是纯扭的，而本文计算结果中则有二阶，分别为第 17、36 阶。大部分情况下，侧弯与扭转振动是耦合在一起的，而且很难区分主从关系。图 13.37～13.41 列出了有代表性的第 1、4、16、17、36 阶振型。

表 13.9 加劲梁侧向弯曲和扭转振动计算结果比较

文献[5]			本文		
阶次	频率(Hz)	振型特点	阶次	频率(Hz)	振型特点
1	0.06826	侧弯 0.5 波，主缆面外	1	0.07080	主跨侧弯 0.5 波
4	0.15849	侧弯 1 波，扭转 1 波	4	0.17025	主跨侧弯、扭转 1 波
10	0.24242	侧弯+扭转，主缆面内面外			
11	0.24775	扭转，主缆面内	16	0.31040	主跨、马湾边跨侧弯并扭转 2 波
13	0.27121	侧弯+扭转，主缆面外	17	0.34070	主跨扭转 0.5 波、主缆面外
18	0.33677	扭转+侧弯，马湾边缆同相	25	0.40528	青衣边缆、主缆面外同相摆振 2.5 波。主跨侧弯并扭转
21	0.38277	侧弯+扭转	27	0.41751	主缆面外同相摆振 2 波。全跨侧弯并扭转
27	0.40912	侧弯+扭转，主缆面外同相	29	0.44174	主跨、马湾边跨侧弯并扭转 2 波，主缆面外同相
29	0.45106	扭转+侧弯（马湾边跨为主）	35	0.51010	马湾边跨、主跨侧弯 2 波，M2 墩横桥向振动，主缆面外同相
34	0.51958	扭转+侧弯，主缆面内面外			
40	0.64759	扭转+侧弯	36	0.52043	主跨扭转 1 波

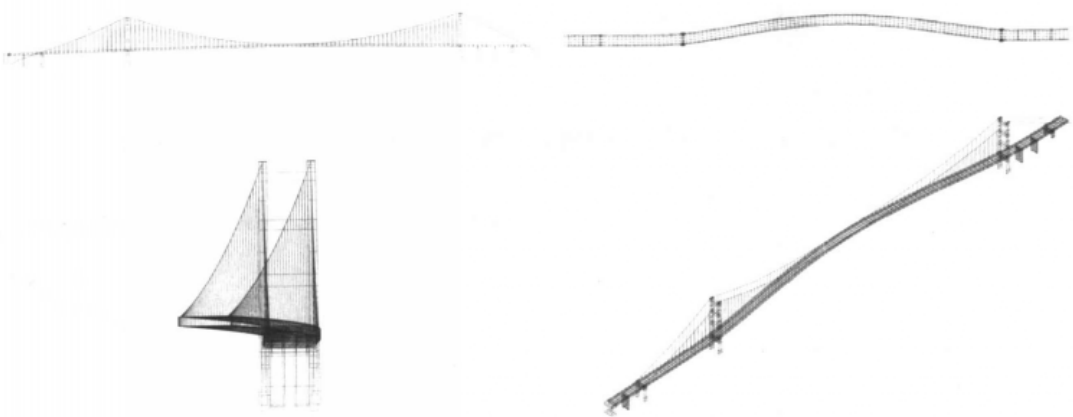


图 13.37 第 1 阶：主跨侧弯、主缆面外 0.5 波（f =0.07080Hz）

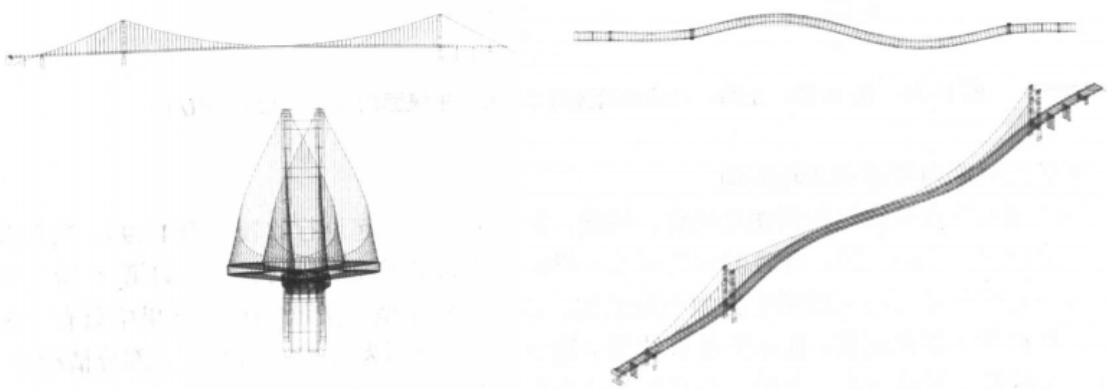


图 13.38 第 4 阶：主跨侧弯、扭转 1 波、主缆面外（f =0.17025Hz）

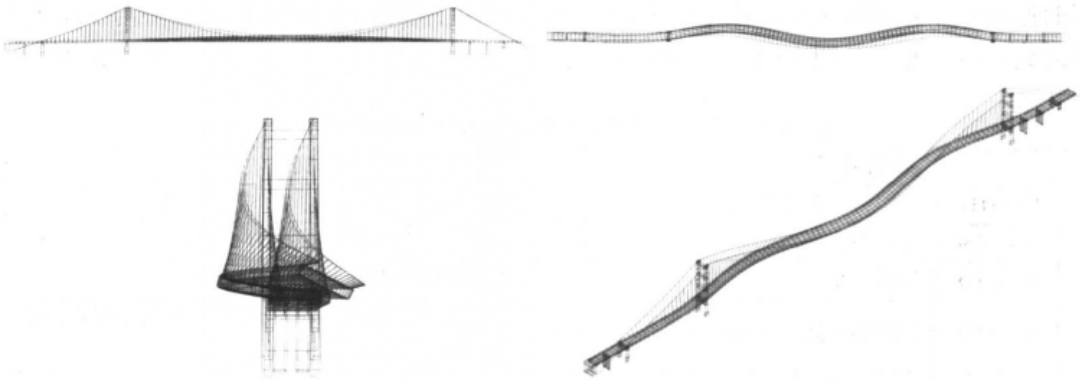


图 13.39　第 16 阶：主跨、马湾边跨侧弯并扭转 2 波、主缆面外（f =0.31040Hz）

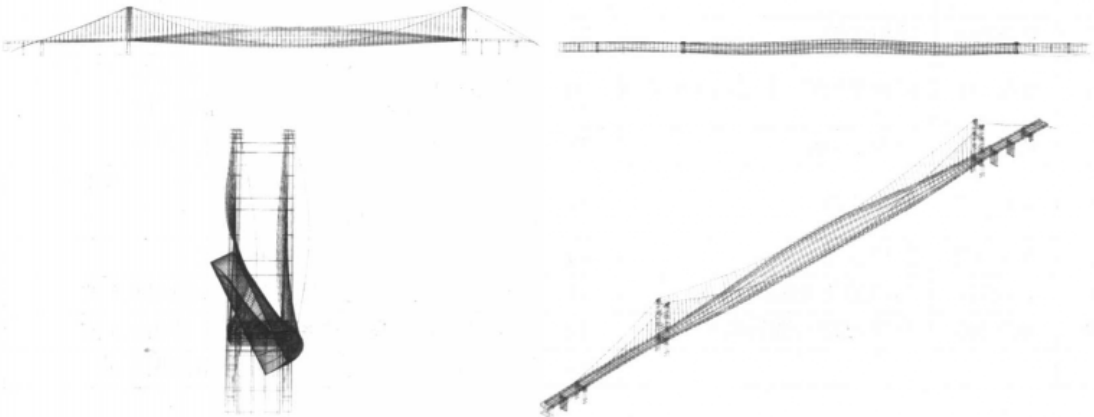


图 13.40　第 17 阶：主跨扭转 0.5 波、主缆面外（f =0.34070Hz）

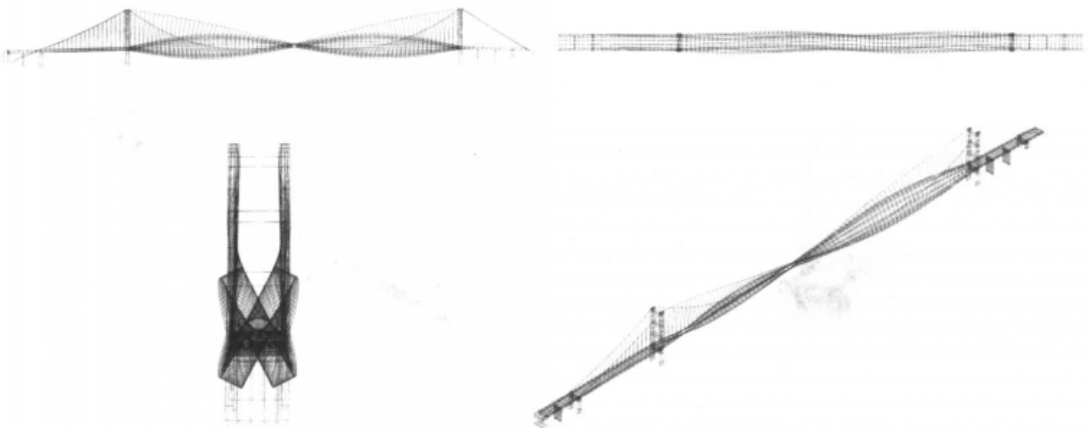


图 13.41　第 36 阶：主跨扭转 1 波（f =0.52043Hz）

③主跨、马湾边跨大缆摆振

主跨大缆摆振振型是成对的，即二根主缆的同相摆振和异相摆振，主缆的摆振总是面外的振动，异相摆振时，一般不会引起加劲梁的振动，但同相摆振时，则往往会使加劲梁产生

侧向弯曲振动并伴有扭转振动，详见表 13.10。马湾边跨主缆由于有吊索，其振型与主跨大缆有类同之处，故归此类。图 13.42～13.46 列出了有代表性的第 8、9、26、33、34 阶振型。

表 13.10 主跨、马湾边跨大缆摆振计算结果比较

文献[5]			本文		
阶次	频率(Hz)	振型特点	阶次	频率(Hz)	振型特点
5	0.19479	异相 0.5 波	6	0.23163	面外异相摆振 0.5 波
7	0.21765	异相 1 波	7	0.23821	面外异相摆振 1 波
8	0.21907	同相 0.5 波，主跨侧弯	8	0.24958	马湾、主跨大缆面外同相摆振 0.5 波、梁侧弯并扭转
9	0.22096	同相 1 波，主跨扭转并侧弯	9	0.25151	面外同相 1 波、主跨侧弯并扭转
15	0.32280	异相 1.5 波	22	0.38604	面外异相摆振 1.5 波
20	0.37242	同相 2 波，青衣塔摆振	33	0.49321	面外同相 2 波，青衣塔横桥向摆振，马湾跨、主跨侧弯并扭转
23	0.38798	异相 2 波	26	0.41714	面外异相摆振 2 波
30	0.46351	同相 2.5 波，主塔同相摆振	40	0.55212	主跨侧弯并扭转 2.5 波，青衣塔横桥向振动，主缆面外同相
31	0.47871	异相 2.5 波	38	0.53020	主缆面外异相摆振 2.5 波
36	0.55672	同相 3 波	42	0.56775	面外同相 3 波，主跨、马湾跨侧弯并扭转 3 波，
37	0.56162	异相 3 波	44	0.59364	面外异相摆振 3 波
16	0.32971	马湾边跨主缆同相 0.5 波	11	0.27464	马湾边跨主缆面外同相 0.5 波
19	0.34163	马湾边跨主缆异相 0.5 波	12	0.27920	马湾边跨主缆面外异相 0.5 波
			34	0.50717	马湾边缆面外异相摆振 1 波

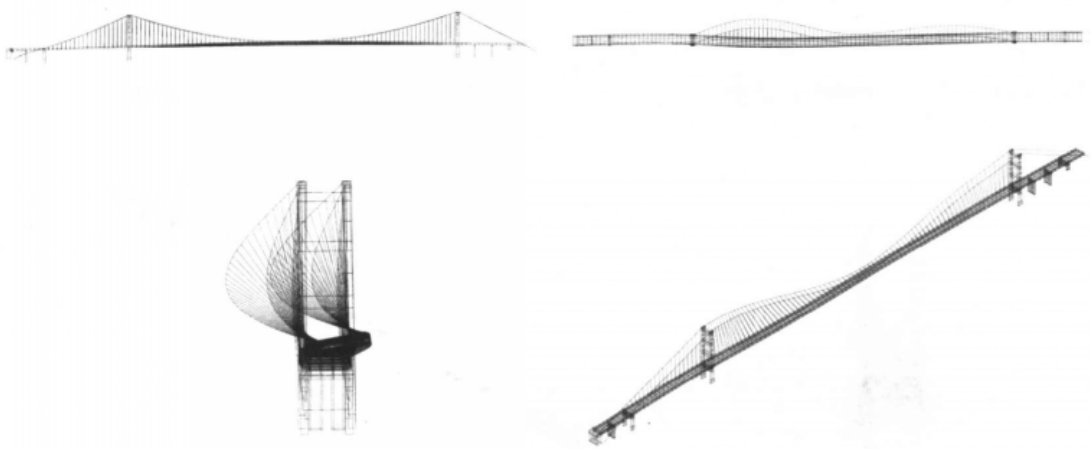


图 13.42 第 8 阶：马湾、主跨大缆面外同相摆振 0.5 波、梁侧弯并扭转（f =0.24958Hz）



图 13.43　第 9 阶：主缆面外同相 1 波、主跨侧弯并扭转（f =0.25151Hz）

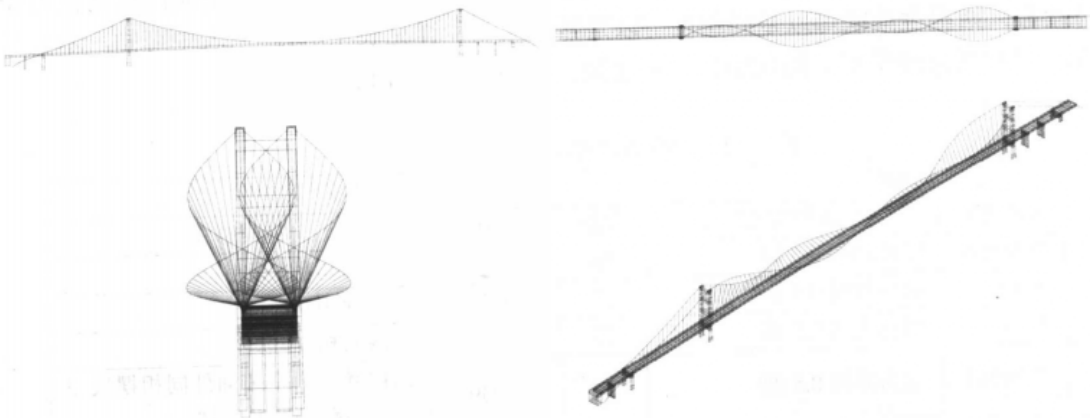


图 13.44　第 26 阶：主缆面外相摆振 2 波（f =0.41714Hz）

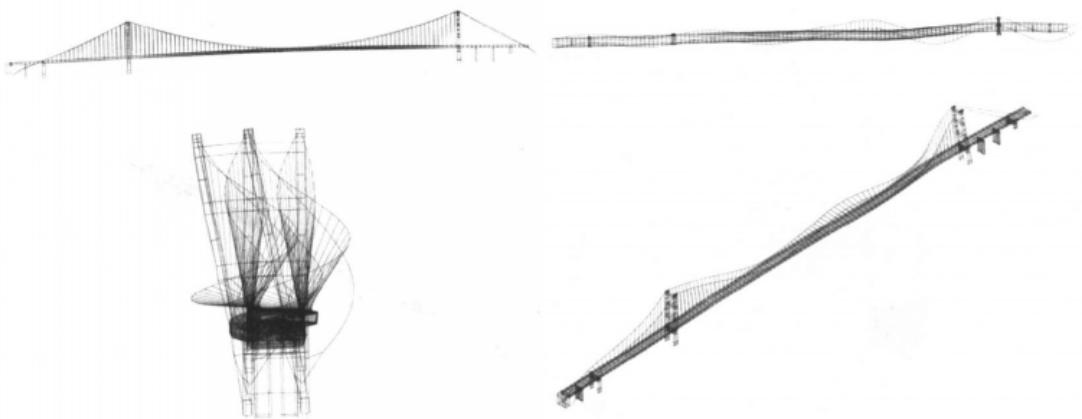


图 13.45　第 33 阶：青衣塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转（f =0.49321Hz）

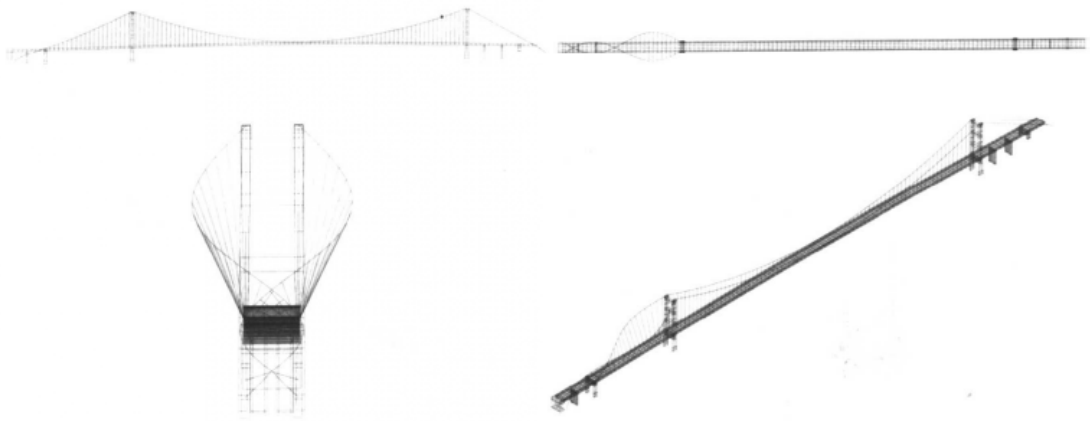


图 13.46 第 34 阶：马湾边缆面外异相摆振 1 波（f =0.50717Hz）

④青衣边跨主缆振动

青衣边跨主缆与马湾边跨的不同，没有吊索，是空缆，其振型既可能是面内振动也可能是面外摆振，两根主缆既可以是同相也可以是异相振动，详见表 13.11。从表中可知，本文与文献[5]的计算结果较接近。图 13.47～13.48 列出了第 21、24 阶振型。

表 13.11 青衣边跨主缆振动计算结果比较

文献[5]			本文		
阶次	频率(Hz)	振型特点	阶次	频率(Hz)	振型特点
22	0.38565	面外同相 0.5 波	18	0.35150	面外同相 0.5 波
24	0.39434	面外异相 0.5 波	21	0.36121	面外异相 0.5
25	0.39825	面内同相 0.5 波	23	0.40321	面内同相 0.5 波
26	0.39903	面内异相 0.5 波	24	0.40395	面内异相、主缆面外同相摆振 2 波、主跨侧弯并扭转

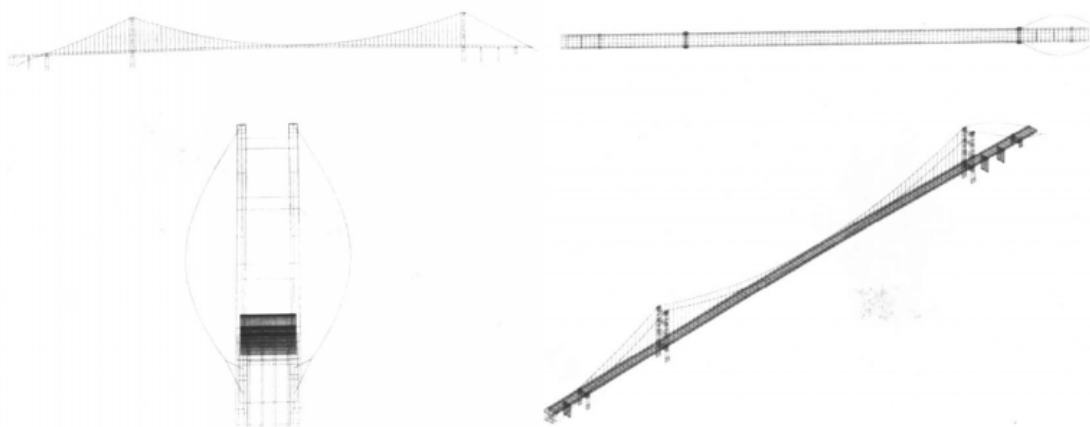


图 13.47 第 21 阶：青衣边缆面外异相摆振 0.5 波（f =0.36121Hz）

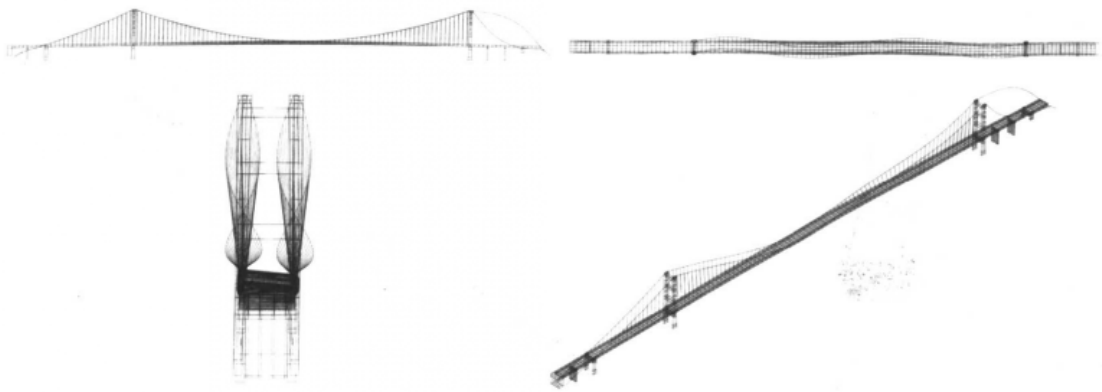


图 13.48　第 24 阶：青衣边缆面内异相、主缆面外同相摆振 2 波、主跨侧弯并扭转（f =0.40395Hz）

⑤以塔为主的振动

以塔为主的振动主要有横桥向的摆振和纵桥向的弯曲振动，主塔可能单独振动，或以一主塔为主的振动，或两主塔一起振动（同相或异相）。前 44 阶振型中以塔为主的振型见表 13.12 所示。图 13.49～13.52 列出了第 33、37、41、43 阶振型。

表 13.12　以塔为主的振动计算结果比较

文献[5]			本文		
阶次	频率(Hz)	振型特点	阶次	频率(Hz)	振型特点
32	0.50118	异相摆振(马湾塔为主)	31	0.46859	马湾塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转
			33	0.49321	青衣塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转
35	0.52672	同相摆振	37	0.52583	马湾塔横桥向摆振，马湾跨、主跨侧弯并扭转，主缆面外同相
38	0.58439	异相摆振			
41	0.64946	同相弯曲振动，马湾塔为主	41	0.56426	主塔纵桥向同相弯曲振动
			43	0.58973	主塔纵桥向异相弯曲振动

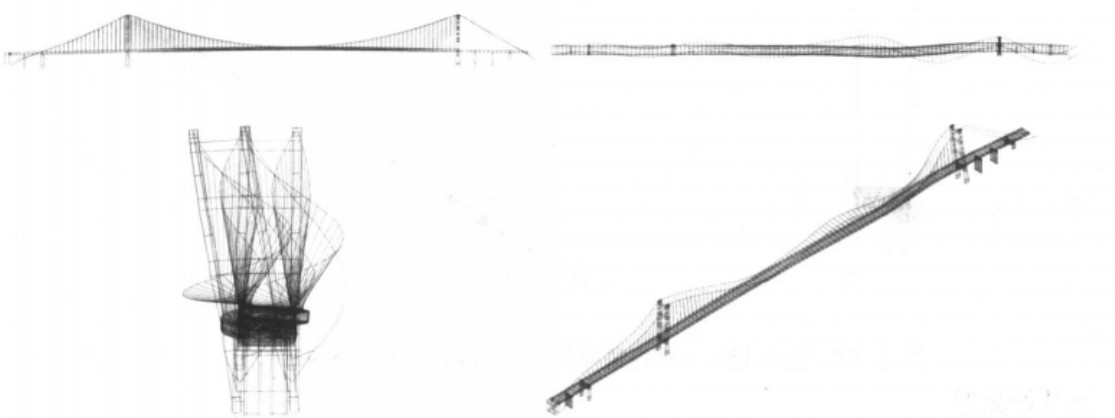


图 13.49　第 33 阶：青衣塔横桥向摆振，主缆面外同相，马湾跨、主跨侧弯并扭转（f =0.49321Hz）

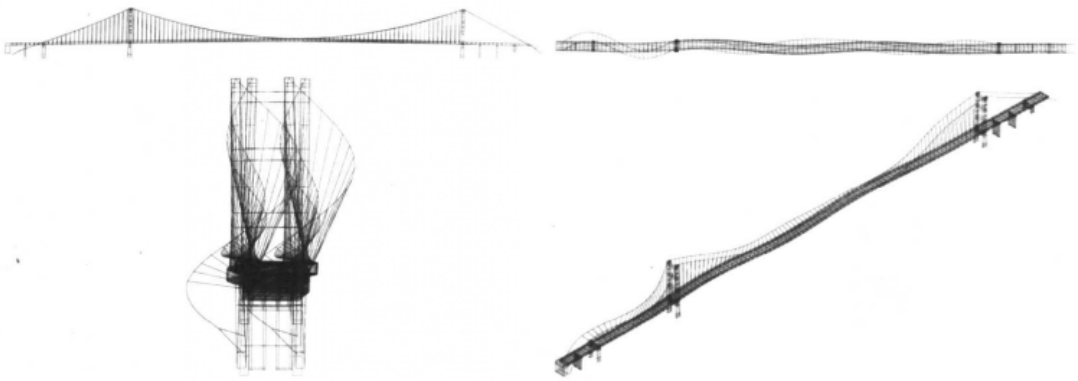


图 13.50 第 37 阶：马湾跨、主跨侧弯并扭转，马湾塔横桥向摆振，主缆面外同相（f =0.52583Hz）

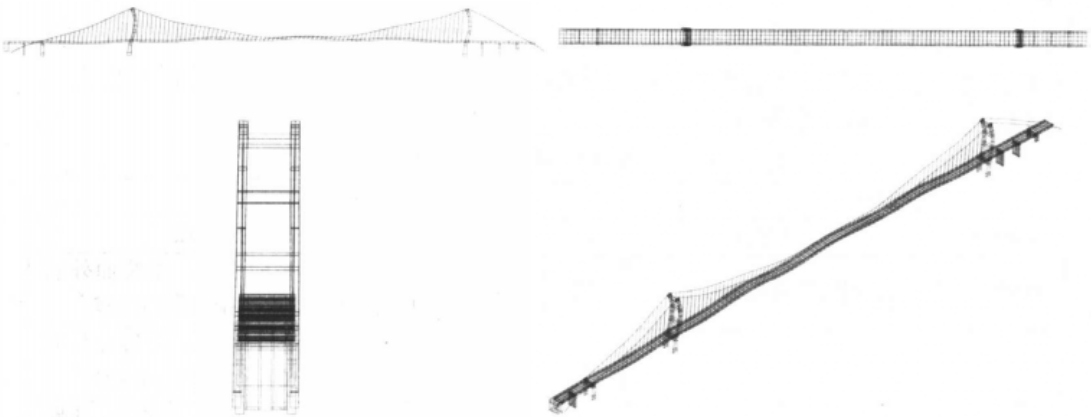


图 13.51 第 41 阶：主塔纵桥向同相弯曲振动（f =0.56426Hz）

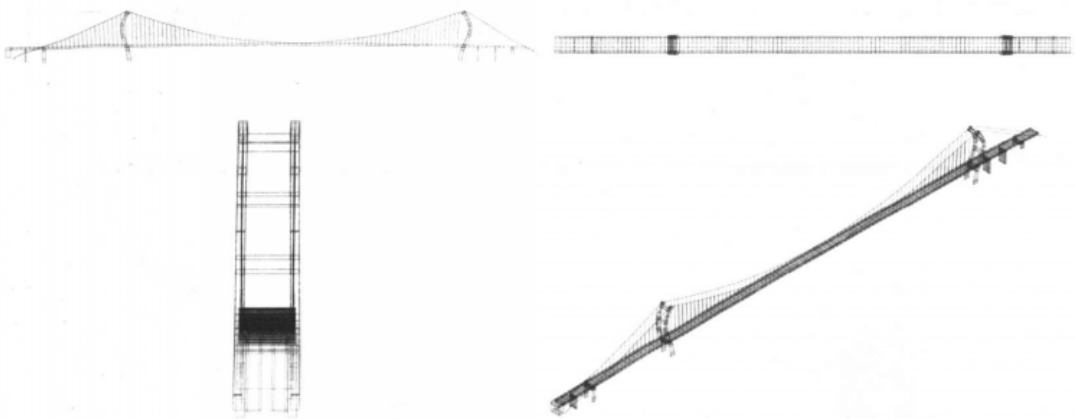


图 13.52 第 43 阶：主塔纵桥向异相弯曲振动（f =0.58973Hz）

⑥其他振型

以桥墩为主的振型：文献[5]中对此类振型并无报道，说明其计算模型中可能没有桥墩，本文前 44 阶计算中则包含了此类振动，分别为第 13、15、20 和 32 阶振动，详见表 13.7，图

13.53 为第 13 阶振型。另外，本文还计算到加劲梁纵桥向拉伸压缩振型，为第 30 阶，详见图 13.54 所示。

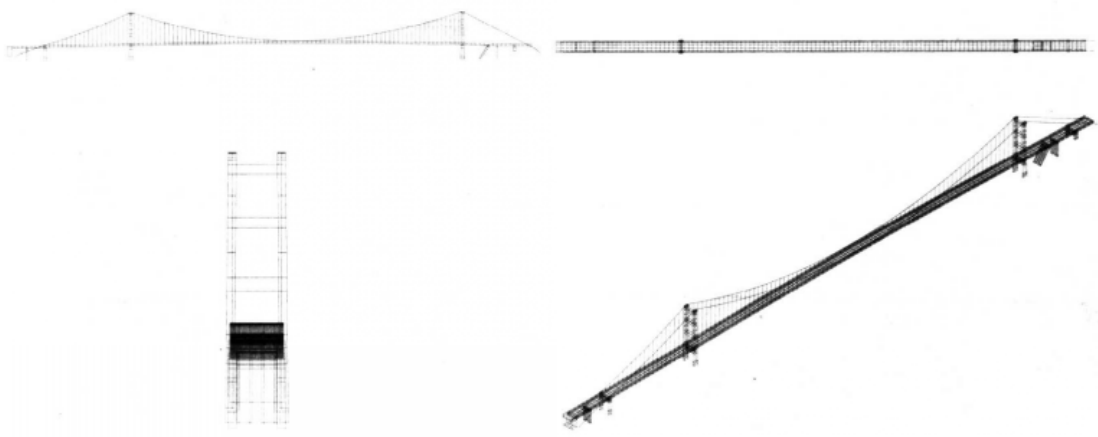


图 13.53　第 13 阶：T3 墩纵桥向弯曲振动（f =0.28855Hz）

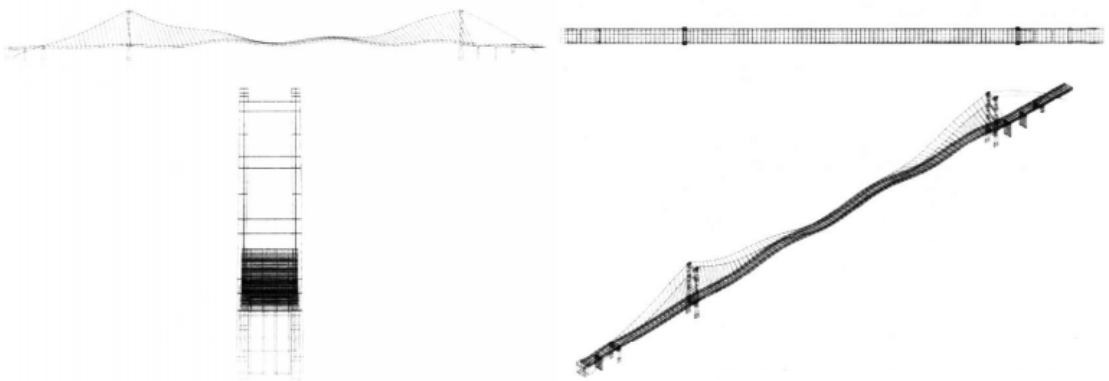


图 13.54　第 30 阶：加劲梁纵漂、竖弯（f =0.45079Hz）

(3)计算频率与实测频率的分类比较

青马大桥在建桥时就在桥梁的不同位置预埋了若干传感器，分别为单向加速度计（*Uniaxial Accelerometer*）15 只，双向加速度计（*Biaxial Accelerometer*）3 只，三向加速度计（*Triaxial Accelerometer*）1 只，风速计（*Anemometer*）6 只，位移传感器（*Displacement Transducer*）2 只，水平感应计（*Level Sensing Station*）10 只，应变计（*Linear Strain Gauge*）106 只等（详见图 13.55 所示），以监视桥梁在运营状态下的安全性。

表 13.13 为香港路政署在青马大桥实测到的振动频率值。表 13.14～13.16 分别列出了侧向弯曲、竖向弯曲和扭转振型的文献[5]和本文计算值与实测值的比较结果，从这些比较中可以看出，本文的计算值更接近于实测值。本文由于采用了一致质量矩阵代替集中质量阵，而且所用的组合单元是协调单元，因此计算频率值应该是结构真实频率的上界[5]。从表 13.16 可知，扭转频率与实测值相比误差较大，但是文献[5]的前 44 阶计算振型中只有一阶是主跨纯扭振动，本文也只有二阶，而实测值有四阶，可见实测的这四阶扭振频率可能并非主跨的纯扭振动频率，很可能是主跨侧弯并扭转振动，导致计算值与实测值比较时的误差较大。如果用带有扭

转的计算侧弯振型频率与实测值作比较，误差也是不大的，详见表 13.17。

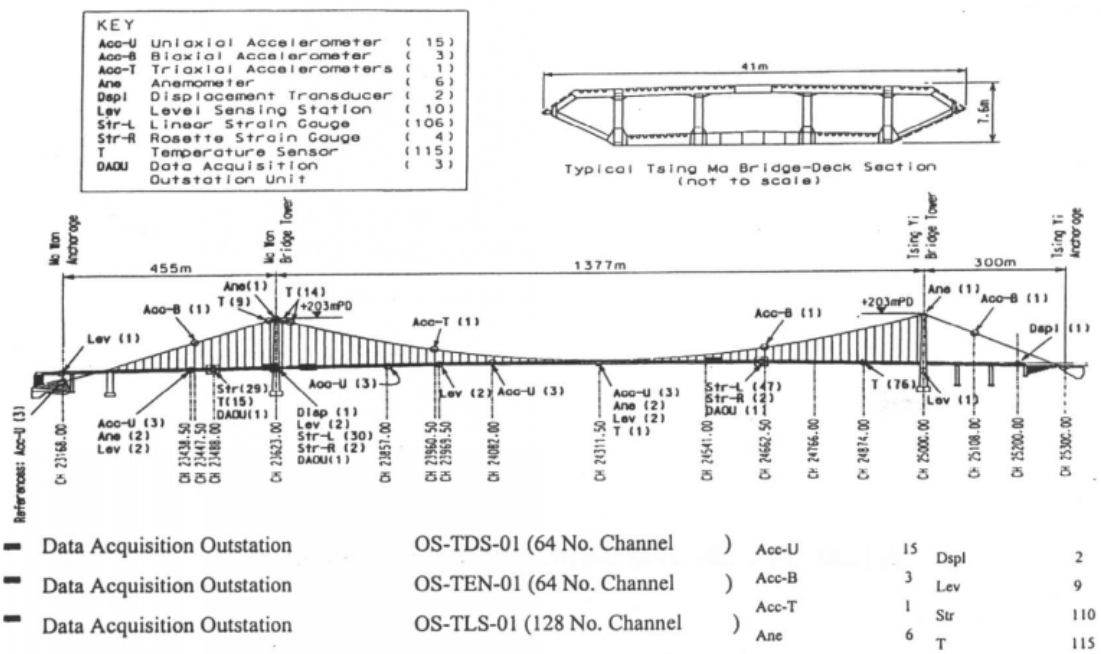


图 13.55 青马大桥传感器安装位置示意图

表 13.13 青马大桥实测振动频率（Hz）（香港路政署）

阶次	侧向弯曲	竖向弯曲	扭转
1	0.070	0.114	0.270
2	0.170	0.133	0.324
3	0.254	0.187	0.486
4	0.301	0.249	0.587

表 13.14 侧弯振动计算频率与实测频率的比较

实测(Hz)	文献[5] (Hz)	阶次	误差	计算值(Hz)	阶次	误差
0.070	0.0683	1	-2.43%	0.07080	1	1.14%
0.170	0.1585	4	-6.76%	0.17025	4	0.15%
0.254	0.2424	10	-4.57%	0.25151	9	-0.98%
0.301	0.2712	13	-9.90%	0.31040	16	3.12%

表 13.15 竖弯振动计算频率与实测频率的比较

实测(Hz)	文献[5] (Hz)	阶次	误差	计算值(Hz)	阶次	误差
0.114	0.1193	2	4.65%	0.11937	2	4.71%
0.133	0.1439	3	8.20%	0.14013	3	5.36%
0.187	0.2003	6	7.11%	0.18992	5	1.56%
0.249	0.2526	12	1.45%	0.25329	10	1.72%

表 13.16　扭转振动计算频率与实测频率的比较

实测(Hz)	文献[5] (Hz)	阶次	误差	计算值(Hz)	阶次	误差
0.270	0.2478	11	-8.22%	0.24958	8	-7.56%
0.324	0.3368	18	3.95%	0.34070	17	5.15%
0.486	0.5196	34	6.91%	0.52043	36	7.08%
0.587	0.6476	40	10.32%	0.55212	40	-5.94%

表 13.17　扭转振动计算频率与实测频率的比较

实测(Hz)	计算值(Hz)	阶次	误差
0.270	0.25150	11	-6.85%
0.324	0.31040	16	-4.20%
0.486	0.49321	33	1.48%
0.587	0.56775	42	-3.28%

13.4　小结

通过以上三个例子，说明改进的实体退化单元（组合单元）保留了有限元法所具有的灵活性与程序统一性特点，适用于复杂结构空间总体分析。不仅大规模地节省了计算工作量，而且仍然能够仔细反映悬索桥各主要构件的力学特征、几何尺寸、物理常数、所在空间位置以及其对系统刚度、质量的贡献，实现了用普通微机进行复杂结构的空间计算分析。

参考文献

[1] 张金江. 斜弯桥的虚拟层合单元法有限元分析. 浙江大学硕士学位论文, 1998

[2] 凌道盛, 张金江, 项贻强, 徐兴.虚拟层合单元法及其在桥梁工程中的应用. 土木工程学报, 1998，31

[3] 徐兴, 凌道盛. 实体退化单元系列. 固体力学学报, 2001，22（计算力学专辑）

[4] 蔡金标. 大跨度悬索桥空间分析的组合单元法. 浙江大学博士学位论文, 2002

[5] Finite element modelling of the TSING MA bridge, Chapter 1 of report No.3 of WASHMS Task7.12, Department of civil & engineering, The Hong Kong Polytechnic University

[6] 丁皓江, 谢贻权, 何福保, 徐兴.弹性和塑性力学中的有限单元法. 北京: 机械工业出版社, 1989

第14章

有限单元法在土坡稳定分析中的应用

土坡系指具有倾斜面的土体。由于坡面倾斜，在本身重量及其他外力作用下，整个土体有从高处向低处滑动的趋势。如果土体内部某个面上的滑动力超过土体抵抗滑动的能力就会发生滑坡。滑坡是一种常见的地质灾害形式，是全球性三大地质灾害之一，每年都给人类带来巨大的生命和财产损失。王恭先统计了上个世纪世界上的一些重大滑坡灾害的实例[1]，一些严重的滑坡造成了成千上万人死亡。因此，土坡稳定性研究具有十分重要的现实意义。本章先介绍关于边坡及其稳定性的一些基本概念，以及在边坡稳定分析中广泛运用的条分法，然后重点介绍有限单元法在土坡稳定分析中的应用。

14.1 边坡及其稳定性的基本概念

本节介绍边坡及其稳定性的一些基本概念，具体内容可参见边坡工程的相关专著。

一、边坡的构成要素及分类

边坡横断面外形和各部位名称如图 14.1 示，坡底、坡趾（坡脚）、坡面、坡肩和坡顶是构成边坡的要素，坡高和坡角是边坡的两个重要几何参数。

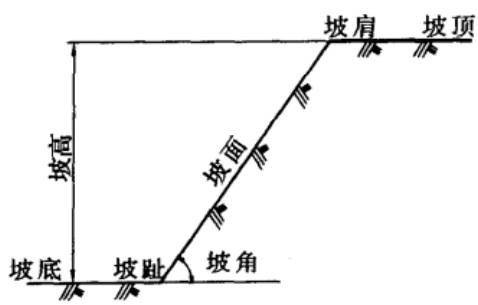


图14.1　边坡构成要素

边坡的分类通常有如下几种方法[2]：

(1) 按照边坡的成因可分为天然边坡和人工边坡。天然边坡是自然形成的山坡和江河湖海的岸坡。人工边坡又可分成开挖边坡和填筑边坡两种。

(2) 按照构成边坡坡体的材料性质可分为粘性土类边坡、碎石类边坡、黄土类边坡和岩石类边坡。

(3) 按照边坡的稳定程度可分为稳定边坡、基本稳定边坡、欠稳定边坡和不稳定边坡。这种分类方法一般根据边坡的稳定性系数的大小进行划分，但无严格的规定。

(4) 按照边坡的高度分类，边坡大于 15m 称为高边坡，小于 15m 称为一般边坡。

(5) 根据边坡的断面形式可分为直立式边坡、倾斜式边坡和台阶形边坡，如图 14.2 所示。根据这三种形式构成复合形式的边坡，如图 14.3 所示。

(6) 根据使用年限分为临时性边坡和永久性边坡。临时性边坡是指工作年限不超过两年的

边坡；永久性边坡是指工作年限超过两年的边坡。

除了上述分类方法外，边坡还可以根据支护结构形式进行分类。

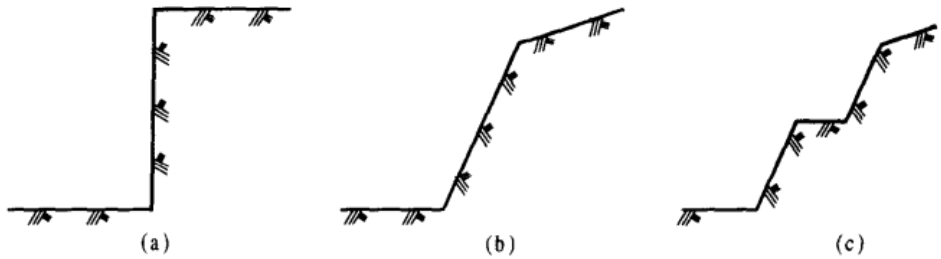


图 14.2　边坡基本形态

(a)直立式边坡　(b)倾斜式边坡　(c)台阶式边坡

二、边坡稳定分析的理论基础

在土力学中，边坡稳定分析是和另外两个分支，即土压力和地基承载力同时发展起来的。1776 年，法国工程师库仑（C.A.Coulomb）提出了计算挡土墙压力的方法，标志着土力学雏形的产生。1857 年，朗肯（W.J.M. Rankine）在假设墙后土体各点处于极限平衡状态的基础上，建立了计算主动和被动土压力的方法。库仑和朗肯在分析土压力时采用的方法后来被推广到地基承载力和边坡稳定分析中，形成了一个体系，这就是极限平衡方法。

极限平衡方法是建立在大家所熟悉的摩尔－库仑强度准则基础上的，其表达式为

$$\tau_f = c' + \sigma' \tan\phi' = c' + (\sigma - p)\tan\phi' \tag{14.1}$$

式中，τ_f 为破坏面上的抗剪强度；c' 为土的有效粘聚力；ϕ' 为土的有效内摩擦角；σ 和 σ' 分别为破坏面上法向总应力和有效应力；p 为孔隙水压力。注意本章中正应力规定以压为正，以符合土力学的习惯。

极限平衡方法的基本特点是，只考虑静力平衡条件和土的摩尔－库仑破坏准则。也就是说，通过分析土体在破坏时的力的平衡来求得问题的解。

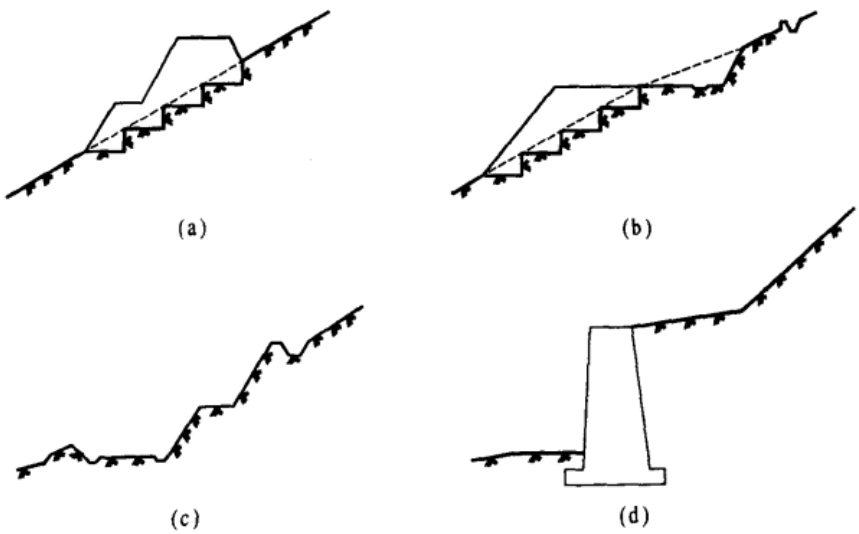


图 14.3 复合边坡形态

三、滑动面及稳定安全系数

土坡稳定问题一般都假定为平面应变问题。彼德森(K.E.Pettterson)、费伦纽斯(W.Fellenius)等在研究边坡问题时都假定滑动面是一个圆柱面，土坡的安全系数是用滑动面上全部抗滑力矩和滑动力矩之比定义的。但实际上，即使是简单均质的土坡，其滑动面也不是圆柱面。因此，圆弧滑动面及其稳定安全系数的定义存在一定的局限性。1955 年，毕肖普（A.W.Bishop）等[3]将滑动面稳定安全系数 F 定义为沿整个滑动面的抗剪强度 τ_f 和实际产生剪应力 τ 的比值。即

$$F=\frac{\tau_f}{\tau} \tag{14.2}$$

毕肖普关于稳定安全系数的定义不仅物理意义更加明确，而且使得该定义适用范围更广，为其后非圆弧滑动分析提供了有利条件[4]。

四、边坡稳定分析方法

边坡工程稳定性分析的方法主要有极限平衡法、塑性极限分析法和有限单元法。

从库仑和朗肯两种不同思路出发，极限平衡方法也逐渐形成两个独立的分支。一是通过研究滑动面上作用的力的静力平衡和确定临界滑动面求得问题的解，在此基础上导出了目前广泛采用的稳定分析的条分法；另一是在假定土体处处达到极限平衡状态的前提下，用特征线法求解应力场，在一定的简化条件下，获得问题的闭合解。遗憾的是，这两种方法都没能很好地考虑边坡的应力应变关系，无法探索边坡变形和稳定的内在联系。而有限单元法则能克服这些不足，将有限单元法应用于边坡稳定分析具有较广阔的应用前景。

（1）极限平衡分析的垂直条分法

首先假定一个扰动因素，使土体从目前的稳定状态进入极限平衡状态。此时，滑体内出现一假想的滑动面。在该滑动面上，每一点的法向应力和切向应力都满足摩尔-库仑强度准则。在边坡稳定问题中，产生这个扰动因素的通常做法是将土体的抗剪强度指标 c' 和 ϕ' 减少为 c'/F 和 $\arctan(\tan\phi'/F)$，称 F 为安全系数。也有学者提出了另一种做法，即在滑动土体每个单元加上一水平体积力，使土体在假设的滑动面上达到极限平衡状态。

在滑动面为一特定的形状时，如直线或对数螺旋线，则可以通过求解静力平衡方程唯一地确定相应该滑动面的上述扰动因素的量值。如果滑动面为任意形状，那么，为了确定沿滑动面的应力分布，需要将滑动土体分成若干垂直土条，通过分析作用于土条上的力来建立平衡方程。为了使问题变得静定可解，需要对土条间的作用力作一些假定，最终获得使该滑动面处于极限平衡状态所需要的扰动因素的量值。

在求得一个滑动面的扰动因素的量值后，还要对不同的滑动面重复这一计算步骤，获得一系列的扰动量。真实的临界滑动面相应于最小的扰动因素量值。

（2）滑移线法

上述极限平衡的垂直条分法获得的是一个满足静力平衡条件的应力场。同时要求滑动面上每一点的应力状态均在摩尔圆上或以内，但是并不要求滑体内每一点的应力状态均在摩尔圆上，即处于极限平衡状态。因此，所获得的解应小于或等于使边坡发生破坏的真实荷载。在塑性力学领域，属下限解。下面要讨论的滑移线法是假定土的破坏区内各点均达到了极限平衡条件，这样，在破坏区域的每一点，除了可以建立静力平衡条件外，还可以增加一个莫尔-库仑破坏条件，在一定的边界条件下，可以用特征线法求解由此形成的方程组。在一些简

化的边界和土质条件下，可以获得闭合解，解得的特征线就是土力学中滑移线，其中一组就是最危险滑动面。

早在 1977 年潘家铮就提出了滑坡极限分析的两条基本原理，即极小值原理和极大值原理，他认为：(i) 滑坡体如能沿许多个滑面滑动，则失稳时它将沿抵抗力最小的一个滑面破坏(极小值原理)；(ii) 滑坡体的滑面一定时，则滑面上的反力(以及滑坡体内的内力)能自行调整，以发挥最大的抗滑能力(极大值原理)。

从 1981 年开始，孙君实根据潘家铮提出的两条基本原理，在潘家铮指导下建立和发展了一种新型的稳定分析理论：模糊极值理论。其基本方程组实质上和 Morgenstern-Price-陈祖煜的方法是一样的，它同时满足力与整体力矩平衡条件。

（3）有限元分析法

建立在极限平衡理论基础上的各种计算土坡稳定的方法，无法考虑土体内部的应力应变关系，无法分析稳定破坏的发生和发展过程，更无法考虑局部变形对土坡稳定的影响。实践证明，稳定和变形有着十分密切的关系。一个土坡在发生整体破坏之前，往往伴随着相当大的垂直沉降和侧向变形。利用有限元法，考虑到土的非线性应力应变关系，求得每一个计算单元的应力及变形后，便可根据不同强度指标确定破坏区的位置及破坏范围的扩展情况。

根据应力、应变的分析结果，将局部破坏与整体破坏联系起来，求得合适的临界滑动面位置，再根据力的平衡关系推得安全系数，是目前利用有限单元法评价边坡稳定性的最常用方法，主要有：

(i) 滑动面搜索法。通过有限元分析得出每个节点的应力张量，假定一个滑动面，即可用有限元数据给出滑动面一点的法向应力 σ_n' 和剪应力 τ。根据莫尔-库仑准则可以求得该点的抗滑力，从而求得滑动面上每个节点的下滑力和抗滑力，再对滑动面上的下滑力和抗滑力进行积分，就可求得到该滑动面的稳定安全系数 F

$$F=\frac{\int_l \tau_f \mathrm{d}l}{\int_l \tau \mathrm{d}l} \tag{14.3}$$

式中，l 为滑动面长度。不断变更滑动面，就可以搜索到一个最小的安全系数，即为边坡稳定安全系数 F_s。

(ii) 有限元强度折减法。将强度折减后代入分析，一直计算到不收敛为止，此时的折减系数即为边坡的稳定安全系数。该方法虽然计算工作量较大，而且受到土体剪应变计算精度和剪应变发展不规律的限制，但具有无需假设滑动面，能很“自然”地获得最危险滑动面的位置等优点，是一种应用前景广阔的分析方法。

14.2　土坡稳定分析的条分法简介

条分法以极限平衡理论为基础，由瑞典人彼得森在 1916 年提出，20 世纪 30～40 年代经过费伦纽斯和泰勒(D.W.Taylor)等人的不断改进，直至 1954 年简布(N.Janbu)提出了普遍条分法的基本原理，1955 年毕肖普明确了土坡稳定安全系数，使该方法在目前的工程界成为普遍采用的方法。

条分法实际上是一个刚体极限平衡分析法。其基本思路是：假定边坡的土体破坏是由于边坡内产生了滑动面，部分坡体沿滑动面滑动造成的。滑动面上的土体服从破坏条件。假设

滑动面已知，通过考虑滑动面形成的隔离体的静力平衡，确定沿滑面发生滑动时的破坏荷载，或者说判断滑动面上的滑体的稳定状态或稳定程度。该滑动面是人为确定的，其形状可以是平面、圆柱面、对数螺旋面或其他不规则曲面。隔离体的静力平衡可以是滑面上力的平衡或力矩的平衡。隔离体可以是一个整体，也可由若干人为分隔的竖向土条组成。由于滑动面是人为假定的，我们只有通过系统地求出一系列滑面发生滑动时的扰动因素的量值，其中，最小量值就是要求的极限值，与之相应的滑动面就是可能存在的最危险滑动面。

条分法的基本假定如下：

把滑动土体竖向分为n个土条，在其中任取一条，记为i。如图 14.4 所示，在该土条上作用的已知力有：土条本身重力W_i，水平作用力Q_i(如地震产生的水平惯性力等)，作用于土条两侧的孔隙水压力p_i及p_{i+1}，作用于土条底部的孔隙水压力p_{bi}。当滑面形状确定后，土条的有关几何尺寸也可确定，如底部坡角α_i，底弧长l_i，滑面上的土体强度c_i'和$\tan\phi_i'$也已确定。要使整个土体达到力的平衡，其未知力还有：土条底部的有效法向法力N_i，共n个；两相邻土条分界面的法向条间力E_i，共$n-1$个；切向条间力X_i，共$n-1$个；两相邻土条间力E_i及X_i合力作用点位置Z_i，共$n-1$个；每一土条底部切力T_i及法向力N_i的合力作用点位置a_i，共n个。另外还有滑体的安全系数F_s，1 个。土条上的力矢多边形如图 14.5 所示。

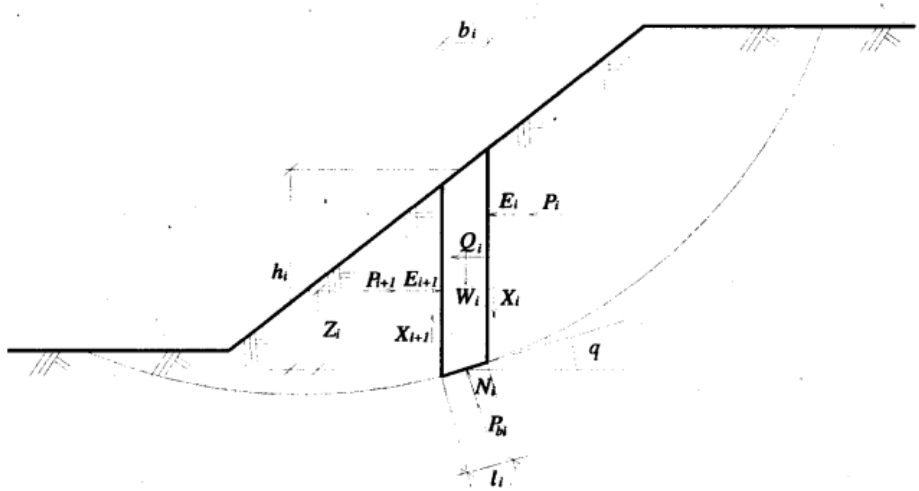


图 14.4 作用在土条上的各种力

综合上述分析，我们得到共计有$5n-2$个未知量，我们能得到的只有各土条水平向及垂直向力的平衡以及土条的力矩平衡共计$3n$个方程。因此，边坡的稳定分析实际上是求解一个高次超静定问题。如果土条比较薄(b_i较小)，T_i和N_i的合力作用点可近似认为在土条底部的中点，a_i变为已知，未知量变为$4n-2$个，与已有的方程数相比，还有$n-2$个未知量无法求出，要使问题有惟一解就必须建立新的条件方程。解决的途径有两个：一个是利用变形协调条件，引进土体的应力应变关系，另一个是作出各种简化假定以减少未知量或增加方程数。前者会使问题变得异常复杂，工程界基本上不采用，后者采用不同的假定和简化，导出不同的方法。

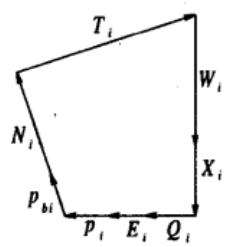


图14.5 土条的力矢多边形

假定$n-1$个X_i值，更简单地假定所有$X_i=0$，这就是常用的毕肖普方法。

假定 X_i 与 E_i 的交角或条间力合力的方向，而有斯宾塞(E.Spencer)法，摩根斯坦-普赖斯(N.R. Morgenstern，V.E . Price)法，沙尔玛(S.K.Sarma)法以及不平衡推力传递法。

假定条间力合力的作用点位置，而有简布提出的普遍条分法。

考虑条间力的作用，可以使稳定安全系数得到提高，但有两点必须注意：一是在土条分界面上不能违反土体破坏准则，即切向条间力得出的平均剪应力应小于分界面土体的平均抗剪强度；二是不允许土条间出现拉应力。如果这两点不能满足，就必须修改原来的假定，或采用别的计算办法。

研究表明，为减少未知量所作的各种假设，在满足合理性要求的条件下，求出的安全系数差别都不大。因此，从工程实用观点来看，在计算方法中无论采用何种假定，并不影响最后求得的稳定安全系数值。我们进行边坡稳定分析的目的，就是要找出所有既满足静力平衡条件，同时又满足合理性要求的安全系数解集。从工程实用角度看，就是找寻安全系数解集中最小的安全系数，这相当于这个解集中的一个点，这个点就是边坡稳定安全系数。

表 14.1 总结了各种极限平衡条分法的主要特点及适用条件。表中 β 为条间力与水平向的夹角。

表 14.1　各种极限平衡条分法的特点及适用条件

方法	静力平衡		对多余未知量的假定	合理性要求	滑动面形状
	力矩	合力			
瑞典法	满　足	部分满足	假定土条间作用力合力平行条底	不校核	圆　弧
余推力法	不满足	满　足	假定土条作用力方向和上土条底平行	不校核	折　线
Bishop 法	满　足	部分满足	假定 $\beta=0$	不校核	圆　弧
Janbu 法	满　足	满　足	假定已知推力线的位置	校　核	任意形状
Spencer 法	满　足	满　足	假定 β 为常数	不校核	任意形状
美国陆军工程师法	不满足	满　足	假定 β 为土坡平均坡度	不校核	任意形状
Morgenstern 法	满　足	满　足	假定 β 为各种函数	校　核	任意形状
Sarma 法	满　足	满　足	对土条切向力作假定	校　核	任意形状

需要说明的是，采用极限平衡法来分析边坡稳定，由于没有考虑土体本身的应力应变关系和实际工作状态，所求出土条之间的内力或土条底部的反力均不能代表边坡在实际工作条件下真正的内力和反力，更不能求出变形。我们只是利用这种通过人为假定的虚拟状态来求出安全系数而已。由于在求解中做了许多假定，不同的假定求出的结果是不相同的，但由于极限平衡法长期在工程中应用，各行业应用不同的方法，都积累了大量的经验，工程界就用这种虚拟状态，来近似模拟实际工作状态，再加上工程经验从而作出工程设计判断。

14.3　利用有限单元法求解滑动面应力的土坡稳定分析

为克服极限平衡法的不足，人们提出了其他数值方法分析边坡的稳定性。有限单元法是其中应用最广的一种数值方法。几乎在开发有限元法的同时，研究者就开始了其与边坡稳定分析中传统的条分法关系的研究。早期的工作是在边坡中定义一个潜在的滑动面，根据有限元法得出的应力分布，计算滑动面上各点的应力水平，然后根据加权平均的原则定义安全系数。他们发现，虽然有限元计算仍处于弹性分析的水平，但按照这一方法计算的安全系数和

极限平衡方法十分接近。

在此基础上，研究者进一步研究在诸多滑动面中寻找一个临界滑动面的可能性。这一工作使用了与极限平衡法相同的计算步骤，同时，也获得了与极限平衡法接近的最小安全系数和临界滑动面。近期研究沿用传统的折减强度指标的做法，通常将非线性有限元数值计算分析收敛条件崩溃作为边坡失稳的判据。此时强度指标被折减的倍数即为安全系数。Griffith(1999)，赵尚毅和郑颖人等(2002)在这方面作出了有益的探索。按照这一方法求解，首先需要具备一个成熟的非线性有限元计算程序。否则，无法判断数值计算发散的原因是边坡系统的物理原因还是有限元程序本身的缺陷。这方面的工作将在下一节介绍。

14.3.1 土体的本构模型

土体的应力应变关系一般是非线性的，在各种应力状态下都有塑性变形；土体受力后有明显的塑性体积变形，而且在剪切时也会引起塑性体积变形(剪胀性)；土体受剪时发生剪应变，其中一部分为弹性剪应变，另一部分与土颗粒间相对错动滑移而产生塑性剪应变，剪应力引起剪应变，体积应力也会引起剪应变；土体还表现硬化和软化特性，应力路径和应力历史对变形有影响，中主应力和固结压力对变形也有影响，而且表现出各向异性。因此，土坡应力场分析是一个复杂的材料非线性问题，第 9 章介绍的材料非线性有限单元法在这里仍然适用。下面只简单介绍土体常用的本构关系。为简单起见，本构方程中不再区分有效应力和总应力。

一、非线性弹性模型

一些土体在特定的加载条件下可以用非线性弹性模型来反映其本构关系，其中最著名的土体非线性弹性模型为邓肯-张的双曲线模型，下面简要介绍这一模型。

(1) 切线弹性模量

邓肯-张的双曲线模型来源于对三轴试验结果的拟合。对于通常的砂土和粘土，Kondner 建议将其应力应变关系用双曲线表示如下：

$$\sigma_1-\sigma_3=\frac{\varepsilon}{a+b\varepsilon}=\frac{1}{a/\varepsilon+b} \tag{14.4}$$

式中，σ_1 为轴向应力（大主应力），σ_3 为围压（小主应力），ε 为轴向应变 ε_a，a、b 为常数。

在式(14.4)中令 $\varepsilon\to\infty$，则得到

$$\frac{1}{b}=(\sigma_1-\sigma_3)_u \tag{14.5}$$

其中，$(\sigma_1-\sigma_3)_u$ 为应力差的渐近值。令抗压强度与应力差渐近值的比值为 R，则有

$$(\sigma_1-\sigma_3)_f=R(\sigma_1-\sigma_3)_u \tag{14.6}$$

式中，$(\sigma_1-\sigma_3)_f$ 为土体抗压强度，R 为破坏比，小于 1，通常为 0.75~1.00。

由式(14.5)式 (14.6)可得到常数 b，即

$$b=\frac{R}{(\sigma_1\text{-}\sigma_3)_f} \tag{14.7}$$

在围压不变的条件下，由式(14.4)求导数，得到土体的切线模量

$$E_T=\frac{\mathrm{d}(\sigma_1-\sigma_3)}{\mathrm{d}\varepsilon}=\frac{a}{(a+b\varepsilon)^2} \tag{14.8}$$

在式(14.8)中令 $\varepsilon=0$，得到

$$E_0=\frac{1}{a} \tag{14.9}$$

E_0 为初始切线模量，常数 a 为初始切线模量的倒数。

将式(14.4)做一些变换，得到

$$\frac{\varepsilon}{\sigma_1-\sigma_3}=a+b\varepsilon \tag{14.10}$$

以 $\dfrac{\varepsilon}{\sigma_1-\sigma_3}$ 为纵坐标，ε 为横坐标，上式是一条直线，a、b 分别是这条直线的截距和斜率。用这种方式整理试验资料，可很方便地确定参数 a 和 b。

试验表明，土体的切线模量随着侧限压力的改变而改变。邓肯-张建议用下式表示初始切线模量与侧限压力之间的关系：

$$E_0=K\ p_a(\frac{\sigma_3}{p_a})^n \tag{14.11}$$

式中，p_a 为大气压力，K、n 为参数。

为了考虑土体的抗拉强度，王复来建议用下式计算初始切线模量，

$$E_0=K\ p_a(\frac{\sigma_3-\sigma_t}{p_a})^n \tag{14.12}$$

其中，

$$\sigma_t=-2c\ \tan(45°\text{-}\phi/2)$$

上式中，σ_t 为土体抗拉强度。

对式(14.12)两边取对数，可知此式在双对数坐标系中是一条直线，利用这一点，可方便地根据试验资料决定参数 K 和 n。

根据 Mohr-Coulomb 破坏准则，抗压强度可由下式表示，

$$(\sigma_1-\sigma_3)_f=\frac{2c\cos\phi+2\sigma_3\sin\phi}{1-\sin\phi} \tag{14.13}$$

把(14.7)和(14.9)式代入(14.8)式，得到切线模量 E_T 如下

$$E_T=\frac{1/E_0}{(\dfrac{1}{E_0}+\dfrac{R\varepsilon}{(\sigma_1-\sigma_3)_f})^2} \tag{14.14}$$

为方便用于有限元计算，从上式中消去应变 ε，把式(14.4)改写为

$$\varepsilon=\frac{a(\sigma_1-\sigma_3)}{1-b(\sigma_1-\sigma_3)}=\frac{\sigma_1-\sigma_3}{E_0(1-\dfrac{R(\sigma_1-\sigma_3)}{(\sigma_1-\sigma_3)_f})}=\frac{\sigma_1-\sigma_3}{E_0(1-R\,S)}$$

将上式代入式(14.14)，消去 ε，得

$$E_T=(1-R\,S)^2E_0 \tag{14.15}$$

其中，

$$S=\frac{\sigma_1-\sigma_3}{(\sigma_1-\sigma_3)_f}$$

把 E_0、S、$(\sigma_1-\sigma_3)_f$ 的表达式代入式(14.15)，有

$$E_T=\left[1-\frac{R(1-\sin\phi)(\sigma_1-\sigma_3)}{2c\cos\phi+2\sigma_3\sin\phi}\right]K\,p_a\left(\frac{\sigma_3-\sigma_t}{p_a}\right)^n \tag{14.16}$$

上式可方便地应用于土体的有限元分析。式中共有 5 个参数，即 c、ϕ、R、K、n，可通过试验确定。

(2) 回弹模量

在实际工程中，可能发生卸荷以及卸荷后再加荷的情况。试验资料表明，土体在卸荷-再加荷过程中，其应力应变关系可用一个统一的切线模量 E_u 表示。卸荷-再加荷的切线模量与应力水平关系不大，而与侧限压力有关，可表示为

$$E_u=K_u p_a\left(\frac{\sigma_3-\sigma_t}{p_a}\right)^n \tag{14.17}$$

式中，E_u 为卸荷-再加荷的切线模量，K_u 和 n 为参数。实际中，此处的 n 值可采用初始切线模量计算式(14.11)中的 n 值，参数 K_u 一般比初始切线模量 E_0 的参数 K 为大。

(3) 切线泊松比

在计算土体的应力和应变时，除了切线模量 E 外，还要用到泊松比。土体的侧向变形和纵向变形之间的关系也可用双曲线表示，

$$\varepsilon_a=\frac{\varepsilon_r}{\nu_0+m\varepsilon_r} \tag{14.18}$$

或

$$\frac{\varepsilon_r}{\varepsilon_a}=\nu_0+m\varepsilon_r \tag{14.19}$$

式中，ε_a 和 ε_r 分别为三轴试验中轴向应变和径向应变，ν_0 为对应于零应变时的初始泊松比，m 为参数。

由式(14.19)可得

$$\varepsilon_r=\frac{\nu_0\varepsilon_a}{1-m\varepsilon_a} \tag{14.20}$$

根据泊松比定义

$$\nu_T=\frac{\mathrm{d}\varepsilon_r}{\mathrm{d}\varepsilon_a} \tag{14.21}$$

即有

$$\nu_T=\frac{\nu_0}{(1-m\varepsilon_a)^2} \tag{14.22}$$

试验资料表明，初始泊松比 ν_0 随侧限压力 σ_3 的增加而减少，可表示如下：

$$\nu_0=g-h\log\left(\frac{\sigma_3-\sigma_t}{p_a}\right) \tag{14.23}$$

式中，g 和 h 为参数。

将式(14.23)代入式(14.22)，得泊松比如下：

$$\nu_T = \frac{g - h\log(\frac{\sigma_3 - \sigma_t}{p_a})}{(1 - m\varepsilon_a)^2} \tag{14.24}$$

上式计算中有三个参数 g、h、m，可由试验确定。

式(14.16)和式(14.24)分别用于计算土体的切线模量和泊松比，是由邓肯提出的，通常称为邓肯模型。

(4) 体积变形模量

1980 年邓肯和 Wong 等人改用体积变形模量 K_T 作为计算参数，定义如下：

$$K_T = \frac{\mathrm{d}\sigma_m}{\mathrm{d}\varepsilon_v} = K_b p_a (\frac{\sigma_3}{p_a})^m \tag{14.25}$$

式中，$\sigma_m = \frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)$ 为平均应力，$\varepsilon_v = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ 为体积应变，K_b 和 m 为由试验确定的常数。

求出 K_T 和 E_T 后，再计算泊松比

$$\nu_T = \frac{3K_T - E_T}{6K_T} \tag{14.26}$$

邓肯模型反映了土体变形的主要规律，但有许多方面没有得到反映。它反映了非线性，把总变形中的塑性变形部分也当着弹性变形处理，通过弹性常数的调整来近似地考虑这部分塑性变形；用于增量计算时，能反映应力路径对变形的影响，通过回弹模量 E_u 与加荷模量 E_T 的差别部分体现加荷历史对变形的影响，但却没反映固结压力增加与降低的差别，也没有反映加荷、卸荷对 ν 的影响；邓肯模型没有反映中主应力对 E、ν 和强度指标的影响，不能反映剪胀性，也不能反映软化和各向异性等问题。

尽管如此，上述邓肯模型由于计算参数是从试验曲线中直接拟合得来的，比较直观，易为工程人员所接受。对于主应力方向没有明显偏转的问题，其计算结果一般是可以接受的。因此，邓肯模型在实际工程计算中得到广泛的应用。

二、双屈服面弹塑性模型

以剑桥模型为代表的弹塑性模型等，从假定的屈服面出发推导出应力应变关系，计算的位移有时偏大一些，但计算结果定性上较为合理。沈珠江院士汲取了邓肯模型和剑桥模型的优点后，提出了双屈服面弹塑性模型，其应力应变关系具有剑桥模型的形式，但有关参数则像邓肯模型一样，可从应力应变关系的试验数据中拟合得到。

（1）屈服函数与弹塑性矩阵

根据第 5 章的弹塑性基本理论，应变增量可分成弹性和塑性应变增量两部分之和。双屈服面弹塑性模型进一步把塑性应变增量分成两部分，即

$$\mathrm{d}\boldsymbol{\varepsilon}^p = \mathrm{d}\boldsymbol{\varepsilon}_1^p + \mathrm{d}\boldsymbol{\varepsilon}_2^p \tag{14.27}$$

假定对应于每一部分塑性应变各有一个屈服面，采用正交流动法则，应变增量可计算如下：

$$\mathrm{d}\boldsymbol{\varepsilon} = \mathrm{d}\boldsymbol{\varepsilon}^e + A_1 \mathrm{d}f_1 \frac{\partial f_1}{\partial \boldsymbol{\sigma}} + A_2 \mathrm{d}f_2 \frac{\partial f_2}{\partial \boldsymbol{\sigma}} \tag{14.28}$$

式中，f_1 和 f_2 分别为两个屈服面函数；A_1 和 A_2 分别相应于屈服面 f_1 和 f_2 的塑性系数。这里，$A\mathrm{d}f$ 相当于第 5 章中的 $\mathrm{d}\lambda$。

记八面体上的剪应力为 τ_m，则

$$\tau_m=\frac{1}{3}\left[(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2+(\sigma_1-\sigma_2)^2\right]^{\frac{1}{2}}$$

沈珠江建议分别用椭圆和幂函数为第 1 和第 2 屈服函数，即

$$f_1=\sigma_m^2+r^2\tau_m^2 \qquad f_2=\frac{\tau_m^s}{\sigma_m} \tag{14.29}$$

式中，r 为椭圆的长、短轴之比，s 为幂次系数。

类似的，记八面体剪切应变为 γ_m，则

$$\gamma_m=\frac{2}{3}[(\varepsilon_2-\varepsilon_3)^2+(\varepsilon_3-\varepsilon_1)^2+(\varepsilon_1-\varepsilon_2)^2]^{\frac{1}{2}}$$

由式(14.28)、式(14.29)两式，得其增量如下：

$$\begin{aligned}\mathrm{d}\varepsilon_v&=\frac{\mathrm{d}\sigma_m}{K}+A_1\frac{\partial f_1}{\partial\sigma_m}\mathrm{d}f_1+A_2\frac{\partial f_2}{\partial\sigma_m}\mathrm{d}f_2\\ \mathrm{d}\gamma_m&=\frac{\mathrm{d}\tau_m}{G}+\frac{2}{3}\left(A_1\frac{\partial f_1}{\partial\tau_m}\mathrm{d}f_1+A_2\frac{\partial f_2}{\partial\tau_m}\mathrm{d}f_2\right)\end{aligned} \tag{14.30}$$

式中，K、G 分别为弹性体积模量和剪切模量。

对(14.29)式求全微分，

$$\begin{aligned}\mathrm{d}f_1&=2\sigma_m\mathrm{d}\sigma_m+2r^2\tau_m\mathrm{d}\tau_m\\ \mathrm{d}f_2&=-\frac{\tau_m^s}{\sigma_{\mathrm{m}}^2}\mathrm{d}\sigma_m+\frac{s\tau_m^{s-1}}{\sigma_m}\mathrm{d}\tau_m\end{aligned}$$

把上式代入式(14.30)，得到

$$\mathrm{d}\varepsilon_v=\frac{\mathrm{d}\sigma_m}{K}+A\mathrm{d}\sigma_m+C\mathrm{d}\tau_m \tag{14.31}$$

$$\mathrm{d}\gamma_m=\frac{\mathrm{d}\tau_m}{G}+\frac{2}{3}(B\mathrm{d}\tau_m+C\mathrm{d}\sigma_m) \tag{14.32}$$

式中，

$$A=4\sigma_m^2A_1+\frac{\tau_m^{2s}}{\sigma_m^4}A_2$$

$$B=4r^2\tau_m^2A_1+\frac{s^2\tau_m^{2s}}{\sigma_m^2\tau_m^2}A_2$$

$$C=4r^2\sigma_m\tau_mA_1-\frac{s\,\tau_m^{2s}}{\sigma_m^3\tau_m}A_2$$

在 π 平面上采用 Prandt-Reuss 流动法则，式(14.32)可扩展为

$$\mathrm{d}e_{ij}=\frac{\mathrm{d}s_{ij}}{2G}+\frac{1}{3}(B\mathrm{d}\tau_m+C\mathrm{d}\sigma_m)\frac{s_{ij}}{\tau_m} \tag{14.33}$$

式中，

$$\mathrm{d}e_{ij}=\mathrm{d}\varepsilon_{ij}-\frac{1}{3}\mathrm{d}\varepsilon_v\delta_{ij}$$

$$s_{ij}=\sigma_{ij}-\sigma_m\delta_{ij}$$

考虑到 $\mathrm{d}\tau_m=\dfrac{1}{3}\dfrac{s_{ij}}{\tau_m}\mathrm{d}s_{ij}$，式(14.33)两边乘上 s_{ij} 后可解出 $\mathrm{d}\tau_m$，再代入式(14.31)和式(14.33)中，得到

$$\begin{aligned}\mathrm{d}\sigma_m&=K_p\mathrm{d}\varepsilon_v-P\frac{1}{\tau_m}\boldsymbol{s}^T\mathrm{d}\boldsymbol{e}\\ \mathrm{d}s_{ij}&=2G\mathrm{d}e_{ij}-P\frac{s_{ij}}{\tau_m}\mathrm{d}\varepsilon_v-Q\frac{s_{ij}}{\tau_m^2}\boldsymbol{s}^T\mathrm{d}\boldsymbol{e}\end{aligned}\tag{14.34}$$

式中，

$$\begin{aligned}K_p&=\frac{K}{1+K\,A}\left(1+\frac{2}{3}\times\frac{K\,G\,C^2}{1+K\,A+G\,D}\right)\\ P&=\frac{2}{3}\times\frac{K\,G\,C}{1+K\,A+G\,D}\\ Q&=\frac{2}{3}\times\frac{G^2\,D}{1+K\,A+G\,D}\\ D&=\frac{2}{3}(B+K\,A\,B-K\,C^2)\end{aligned}$$

对于平面应变问题，式(14.34)变成如下形式：

$$\mathrm{d}\boldsymbol{\sigma}=\boldsymbol{D}_{ep}\mathrm{d}\boldsymbol{\varepsilon}\tag{14.35}$$

式中，

$$\mathrm{d}\boldsymbol{\sigma}=\{\mathrm{d}\sigma_x,\mathrm{d}\sigma_y,\mathrm{d}\tau_{xy}\}^T$$

$$\mathrm{d}\boldsymbol{\varepsilon}=\{\mathrm{d}\varepsilon_x,\mathrm{d}\varepsilon_y,\mathrm{d}\gamma_{xy}\}^T$$

$$\boldsymbol{D}_{ep}=\begin{bmatrix}M_1-P\dfrac{s_x+s_y}{\tau_m}-Q\dfrac{s_x^2}{\tau_m^2} & M_2-P\dfrac{s_x+s_y}{\tau_m}-Q\dfrac{s_xs_y}{\tau_m^2} & -P\dfrac{s_{xy}}{\tau_m}-Q\dfrac{s_{xy}s_x}{\tau_m^2}\\ M_2-P\dfrac{s_x+s_y}{\tau_m}-Q\dfrac{s_xs_y}{\tau_m^2} & M_1-P\dfrac{s_x+s_y}{\tau_m}-Q\dfrac{s_y^2}{\tau_m^2} & -P\dfrac{s_{xy}}{\tau_m}-Q\dfrac{s_{xy}s_y}{\tau_m^2}\\ -P\dfrac{s_{xy}}{\tau_m}-Q\dfrac{s_{xy}s_x}{\tau_m^2} & -P\dfrac{s_{xy}}{\tau_m}-Q\dfrac{s_{xy}s_y}{\tau_m^2} & G-Q\dfrac{s_{xy}^2}{\tau_m^2}\end{bmatrix}\tag{14.36}$$

其中，

$$M_1=K_p+\frac{4G}{3}\qquad M_2=K_p-\frac{2G}{3}$$

（2）塑性系数

假定塑性系数 A_1 和 A_2 只是应力状态的函数，与应力路径无关，于是室内简单应力路径下测得的结果，可以直接应用于现场的复杂应力条件。利用常规三轴试验结果确定塑性系数，

$$\mathrm{d}\gamma_m=\frac{\sqrt{2}}{3}(3\mathrm{d}\varepsilon_1-\mathrm{d}\varepsilon_v)$$

$$\mathrm{d}\sigma_m=\frac{1}{3}\mathrm{d}\sigma_1$$

$$d\tau_m = \frac{\sqrt{2}}{3} d\sigma_1$$

式中，ε_v 为体积应变。将上式代入(14.31)式，并定义

$$E_T = d\sigma_1 / d\varepsilon_1$$

$$\nu_T = d\varepsilon_v / d\varepsilon_1$$

得到，

$$\begin{aligned} \frac{9}{E_T} &= \frac{1}{K} + \frac{3}{G} + 4(\sigma_m + \sqrt{2}r^2\tau_m)^2 A_1 + \frac{\tau_m^{2s}}{\sigma_m^2}\left(\frac{1}{\sigma_m} - \frac{\sqrt{2}s}{\tau_m}\right)^2 A_2 \\ \frac{3\nu_T}{E_T} &= \frac{1}{K} + 4\sigma_m(\sigma_m + \sqrt{2}r^2\tau_m)A_1 + \frac{\tau_m^{2s}}{\sigma_m^2}\left(\frac{1}{\sigma_m} - \frac{\sqrt{2}s}{\tau_m}\right)^2 A_2 \end{aligned} \tag{14.37}$$

由上式可解出 A_1、A_2：

$$A_1 = \frac{\tau_m\left(\frac{9}{E_T} - \frac{3\nu_T}{E_T} - \frac{3}{G}\right) + \sqrt{2}s\sigma_m\left(\frac{3\nu_T}{E_T} - \frac{1}{K}\right)}{4\sqrt{2}\left(\sigma_m + \sqrt{2}r^2\tau_m\right)\left(s\sigma_m^2 + r^2\tau_m^2\right)} \tag{14.38}$$

$$A_2 = \frac{\sigma_m^4\tau_m^2}{\tau_m^{2s}} \times \frac{\sigma_m\left(\frac{9}{E_T} - \frac{3\nu_T}{E_T} - \frac{3}{G}\right) - \sqrt{2}r^2\tau_m\left(\frac{3\nu_T}{E_T} - \frac{1}{K}\right)}{\sqrt{2}\left(\sqrt{2}s\sigma_m - \tau_m\right)\left(s\sigma_m^2 + r^2\tau_m^2\right)} \tag{14.39}$$

上面公式中，E_T 可用式(14.16)计算。

沈珠江用抛物线拟合试验得出的 ε_v □ ε_1 关系曲线，由定义 $\nu_T = d\varepsilon_v / d\varepsilon_1$，得到

$$\nu_T = 2C_d\left(\frac{\sigma_3}{p_a}\right)^d \frac{E_0 R_s}{\sigma_1 - \sigma_3}\frac{1 - R_d}{R_d}\left(1 - \frac{R_s}{1 - R_s}\frac{1 - R_d}{R_d}\right) \tag{14.40}$$

式中，

$$R_d = (\sigma_1 - \sigma_3)_d / (\sigma_1 - \sigma_3)_u$$

式中，C_d 为常数，d 为体应变随 σ_3 变化的幂次；R_d 为最大体应变发生时的应力比(剪胀比)。卸荷-再加荷的切线模量由式(14.17)计算求得。假定泊松比 ν 为常数，弹性体积模量 K 和剪切模量 G 可由下式计算，

$$K = \frac{E_u}{3(1-2\nu)} \qquad G = \frac{E_u}{2(1+\nu)} \tag{14.41}$$

对大多数土体，可取 ν=0.30。

屈服面参数 r 和 s，对于土体，根据沈珠江的计算结果，可取 $r=2$，$s=3$。

14.3.2 安全系数的定义

在极限平衡分析中，整个滑动面的安全系数定义为滑动面上抗剪强度和土体剪应力的比值。在滑动面应力基于有限元分析的稳定分析中，根据沿滑动面剪应力的计算方法，安全系数大致有如下三种定义方法[5]。

（1）基于应力水平的定义法

保持σ_3'不变，作一个与摩尔-库仑强度包线相切的应力圆，称为破坏应力圆，相应的应力圆直径为$(\sigma_1'-\sigma_3')_f$，$(\sigma_1'-\sigma_3')/(\sigma_1'-\sigma_3')_f$为应力水平，表示当前应力圆直径与破坏应力圆直径之比，反映了强度的发挥程度。

整个滑动面的安全系数定义为

$$F_{FE1}=\frac{\int \mathrm{d}l}{\int \frac{\sigma_1'-\sigma_3'}{(\sigma_1'-\sigma_3')_f}\mathrm{d}l} \tag{14.42}$$

（2）基于剪应力的定义法

滑动面上某点的应力状态为$\sigma_x,\sigma_y,\tau_{xy}$，法向应力 σ_n'和切向应力τ可由下式确定：

$$\tau=\frac{1}{2}(\sigma_y-\sigma_x)\sin 2\alpha+\tau_{xy}\cos 2\alpha \tag{14.43}$$

$$\sigma_n'=\sigma_x'\sin^2\alpha+\sigma_y'\cos^2\alpha-\tau_{xy}\sin 2\alpha \tag{14.44}$$

式中，α为此处滑面切线与x轴夹角。

该点的抗剪强度由下式确定：

$$\tau_f=c'+\sigma_n'\tan\phi' \tag{14.45}$$

整个滑面的安全系数定义为

$$F_{FE2}=\frac{\int(c'+\sigma_n'\tan\phi')\mathrm{d}l}{\int\tau\mathrm{d}l} \tag{14.46}$$

（3）基于应力水平加权强度的定义法

Donald 和 Tan 提出了如下公式计算一个滑动面的安全系数方法，

$$F_{FE3}=\frac{\int(c'+\sigma_n'\tan\phi')\mathrm{d}l}{\int\frac{\sigma_1'-\sigma_3'}{(\sigma_1'-\sigma_3')_f}(c'+\sigma_n'\tan\phi')\mathrm{d}l} \tag{14.47}$$

14.3.3　临界滑动面搜索方法

无论是基于极限平衡理论，还是以有限元理论为基础的土坡稳定分析，都要分两步来进行：第一步，对于某一可能的滑动面，分析其滑动和抗滑因数，从而构造安全系数F与潜在滑动面的函数关系；第二步，对许多潜在的滑动面，确定最小安全系数F_s的滑动面。近 20 年来，临界滑动面的搜索问题得到很大的重视，提出了不少确定边坡圆弧或非圆弧临界滑动面的优化搜索技术和方法。这些方法可以分成五大类：变分法、固定模式搜索法、数学规划法、随机搜索法、人工智能法。以下简单介绍这些方法，详细内容可参见相关的综述性文献[6-8]。

1977 年，Baker 和 Garber[9]、Castill 和 Revilla[10]、Revilla 和 Castill[11]、Ramamurthy[12]等尝试着将变分原理应用于边坡稳定分析，他们将滑动面、正应力的分布和条间力的分布都看成变量，将边坡安全系数看成这些变量的泛函，利用变分原理，求得使安全系数泛函F达到极小的临界滑动面和分布力。这一方法是一种解析方法，因此其应用范围存在很大的局限性。De Josselin 和 De Jong[13]、Luceno 和 Castillo[14]等人对变分法理论的正确性和分析问题的局限

性提出了质疑。

固定模式搜索法是搜索点位置和搜索过程在搜索进行之前就已经明确限定的一种搜索方法。区格搜索法、模式搜索法、二分法和单形体映射方法都属于固定模式搜索法。区格搜索法是早期计算机辅助边坡稳定分析中常用的方法。该方法具有原理简单，不受安全系数函数形态影响，不会陷于局部极小值等优点而被广泛应用。为获得最小的安全系数，一般需要设定较大的搜索范围，因此计算量大。区格搜索法一般只用于圆弧滑动面的搜索。模式搜索法、二分法和单形体映射方法在实际应用中各有优点，但有一个共同的缺点，即容易陷于局部极小值。因此，它们的应用受到一定的限制。

数学规划法的基本原理就是借鉴变分法的思想，把滑动面看成一个变量，再把安全系数 F 看成是滑动面的泛函。上个世纪八十年代，国内外学者对此方法研究很多，主要有 Celestino 和 Duncan[15]，Arai 和 Tagyo[16]，Li 和 White[17]，孙君实[18]，Yamagami 和 Ueta[19]，陈祖煜和邵长明[20]等。该方法一般需要对目标函数 F 进行求导运算，以确定下降梯度的方向。数学规划法分析临界滑动面主要有线性-非线性规划法和动态规划法。实践表明，动态规划法是一种比较实用、合理、高效的搜索方法，特别是与有限元方法在边坡稳定分析中的结合使其展现出较大的应用和发展潜力。其中，Yamagami 和 Ueta[19]较早根据非线性有限元法计算出的应力场，按动态规划方法搜索成层土坡内的临界滑动面。其后的工作有 Zhou 和 Williams[21]，史恒通[22]、金小惠[23]的研究等。

随机搜索法分为随机产生方法和随机修改方法。随机产生方法是随机地产生巨大数量的一系列潜在滑动面，分别进行计算比较，并认为其中安全系数最小的滑动面为临界滑动面。随机修改方法是在现有最优解的基础上进行微小的随机修改，然后将新的滑动面和原来的滑动面进行比较，依次搜索下去。一般说来，随机修改方法的搜索效率要比随机产生法方法高。随机搜索法的优点是，对复杂的边坡分析不会遇到数值问题，也不会陷入局部极值，而且理论比较简单，易于编程计算。缺点是计算量大。

目前，在临界滑动面搜索中最常用的人工智能方法是遗传算法。该方法是由 Holland 教授首先提出来的，它是一种模拟达尔文的生物进化理论提出来的一种算法，把生物进化过程中的选择、交叉、变异等概念引入到算法中，是一种全局概率的智能搜索方法。遗传算法已在很多领域得到应用，已被公认为是一种全局搜索能力强、搜索效率高的算法。

14.3.4 算例

例 14-1 简单均质土坡稳定分析[5]

如图 14.5 所示简单均质边坡，材料参数如表 14.2 所示。

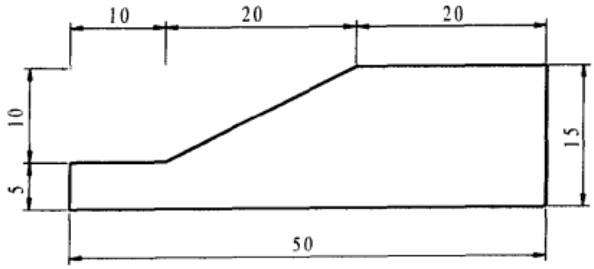


图 14.5 简单均质边坡几何尺寸（单位：米）

表 14.2 材料参数

c(kN/m^2)	ϕ(°)	γ(kN/m^3)	E(kN/m^2)	ν	K_0
3.0	19.6	20.0	10000.0	0.25	0.65

采用不同的极限平衡分析程序进行计算，得到一致的结果，最小安全系数为 1.0。表 14.3 给出了按不同安全系数定义法的有限元法计算结果。从表可以看出，一次填筑和分层填筑边坡的计算值非常接近，开挖边坡的计算值略高。所有计算结果和极限平衡法的解答一致。

表 14.3　安全系数比较

类型	安全系数		
	F_{FE1}	F_{FE2}	F_{FE3}
分层填筑	1.001	1.001	1.001
一次填筑	1.000	1.003	1.000
开　挖	1.082	1.044	1.044

14.4　土坡稳定分析的有限元强度折减法

强度折减法分析边坡稳定安全系数的基本原理是将边坡强度参数粘聚力 c' 和内摩擦角 ϕ' 的正切 $\tan\phi'$ 除以一个折减系数 F，得到一组新的 c' 和 ϕ' 值，然后作为一组新的材料参数输入，再进行试算。使计算不收敛的最小折减系数被称为土坡的最小安全系数。此时，土坡达到极限状态，发生剪切破坏，同时可得到最危险的滑动面。

强度折减法特别适合用有限元方法来实现。早在 1975 年，O.C.Zienkiewicz 就采用此方法分析边坡稳定，只是由于需要花费大量的机时而在具体应用中受到限制[24]。其后，随着微机的发展和有限元计算技术的提高，有限元强度折减法成为边坡稳定分析研究的新趋势，国内外众多学者发表了相关的论文。比较有影响的有宋二祥（1997）[25]，Griffiths 和 Lane(1999)[26] 和郑颖人（2002、2003）[27][28]等人的研究工作。

14.4.1　土体的本构模型

一、土体的模型参数

采用强度折减法分析边坡稳定性时，土体模型主要包括表 14.4 中列出的六个参数。

表 14.4　六个参数的土体模型

c'	ϕ'	ψ	E'	ν'	γ
摩擦力	凝聚力	剪胀角	杨氏模量	泊松比	容量

在土塑性力学中，剪胀角在屈服过程中影响土体的体积变化。众所周知，在屈服过程中，真实的土体体积是变化的。例如一种中等密度的材料在剪切过程中，初始体积是减小的（$\psi<0$），随后是剪胀阶段（$\psi>0$），最终是在体积为常量条件下的屈服（$\psi=0$）。

随之而来的问题是 ψ 应如何取值。在理想弹塑性模型中，一般只考虑剪胀角为常数的情

况。在相关流动法则下，土体的膨胀角等于内摩擦角，即 $\psi=\phi'$。在稳定分析中，研究对象常常是没有约束的天然边坡，因此剪胀角的选择就变得不太重要。这里研究的主要目的是准确地预测边坡的安全系数，Griffiths 等采用的是 $\psi=0$ 的折中值，对应屈服时体积改变量为零，属非关联流动。

参数 c' 和 ϕ' 指土体的有效凝聚力和内摩擦角。弹性参数 E' 和 ν' 是土体的杨氏模量和泊松比。如果已经假定了泊松比（$0.2<\nu'<0.3$），则杨氏模量与在一维固结中得到的土的压缩率有关，

$$E'=\frac{(1+\nu')(1-2\nu')}{m_v(1-\nu')}$$

式中，m_v 是体积压缩率系数。

虽然弹性参数的值对计算破坏前的变形有很大影响，但对边坡稳定分析中安全系数的影响却很小。Griffiths 和 Lane 认为没有 E' 和 ν' 的值时，可以给一个象征性的值（如 $E'=10^5\text{kN/m}^2$，$\nu'=0.3$）。

在计算过程中，土体的总单位体积的容重 γ 与由重力产生的等效节点力是成比例的。

总而言之，在采用有限元强度折减法分析边坡稳定时采用的最重要的参数与传统的方法是一样的，即容重 γ、强度参数 c' 和 ϕ'，以及问题的几何形状。

二、屈服准则

以下两种屈服准则是有限元强度折减法中常用的。

（1）摩尔－库仑准则（M-C 准则）

虽然有许多模拟土的强度的破坏准则，但摩尔－库仑准则仍然是岩土工程实践中应用最广泛的一个。该准则可以写成如下的形式：

$$f(\boldsymbol{\sigma},\boldsymbol{\sigma}^p,\kappa)=\frac{\sigma_1'-\sigma_3'}{2}-\frac{\sigma_1'+\sigma_3'}{2}\sin\phi'-c'\cos\phi' \qquad (14.48)$$

式中，σ_1' 和 σ_3' 分别是最大和最小有效主应力。将上式中的 ϕ' 用 ψ 代替就得到塑性势函数 Q。

传统的极限平衡法采用摩尔－库仑准则，但由于摩尔－库仑准则的屈服面为不规则的六角锥体表面，与 π 平面的交线是不等边六角形，存在尖顶和菱角，给有限元分析带来困难。因此，学者建议采用以下德鲁克－普拉格准则。

（2）德鲁克－普拉格准则（D-P 准则）

Griffiths 等人的研究采用 M-C 准则，其成果受到国内外学者的重视。鉴于摩尔－库仑准则在数值分析中存在的困难，赵尚毅和郑颖人等建议采用德鲁克－普拉格（Drucker-Prager）准则，其屈服函数为

$$F=\alpha I_1+\sqrt{J_2}=\kappa \qquad (14.49)$$

式中，I_1、J_2 分别为应力张量的第一不变量和应力偏张量的第二不变量。不同的 α、κ 在 π 平面上代表不同的圆，如图 14.6 示。

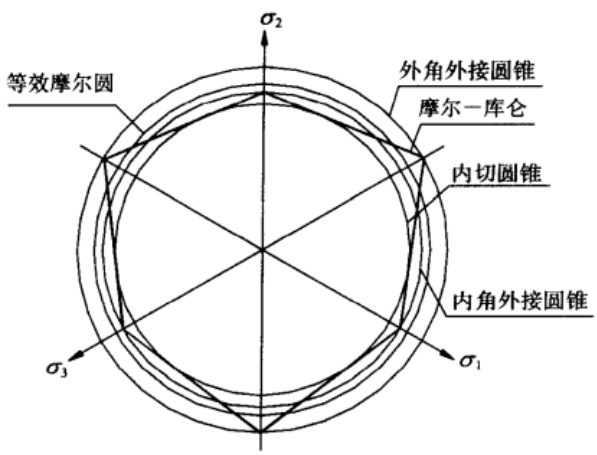


图14.6 各屈服准则在 平面上的曲线

由于摩尔－库仑模型及其参数的确定在岩土工程中积累了丰富的经验，建立 D-P 准则和 M-C 准则间的联系，不仅可以使得 D-P 准则中的参数易于工程确定，也使得 D-P 模型具有更明确的物理意义。根据 π 平面上 M-C 不等边六边形和 D-P 圆的关系，科研人员提出如下参数换算关系。

表 14.5　各准则参数换算表

编号	准则种类	α	κ
DP1	外角点外接 D－P 圆	$\frac{2\sin\phi'}{\sqrt{3}(3-\sin\phi')}$	$\frac{6c'\cos\phi'}{\sqrt{3}(3-\sin\phi')}$
DP2	内角点外接 D－P 圆	$\frac{2\sin\phi'}{\sqrt{3}(3+\sin\phi')}$	$\frac{6c'\cos\phi'}{\sqrt{3}(3+\sin\phi')}$
DP3	内切 D－P 圆	$\frac{\sin\phi'}{\sqrt{3}\sqrt{3+\sin^2\phi'}}$	$\frac{3c'\cos\phi'}{\sqrt{3}\sqrt{3+\sin^2\phi'}}$
DP4	等面积 D－P 圆	$\frac{2\sqrt{3}\sin\phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}}$	$\frac{6\sqrt{3}c'\cos\phi'}{\sqrt{2\sqrt{3}\pi(9-\sin^2\phi')}}$

在非相关流动法则中，将屈服函数中的 α 用 α_ψ 来代替就得到塑性势函数 Q。其中，α_ψ 为剪胀角 ψ 的函数。

增量形式的应力应变关系已经在第 5 章中给出详细推导，读者可参阅相关章节。

14.4.2　安全系数的确定和破坏的定义

一个土质边坡的安全系数是这样定义的，为了使边坡达到破坏，给原始的剪切强度参数除以一个数，这个数就是安全系数（安全系数的这种定义和极限平衡法中的定义是相同的）。则经过折减的剪切强度参数 c' 和 ϕ' 变为

$$c'_f = c'/F \tag{14.50}$$

$$\phi'_f = \arctan\left(\frac{\tan\phi'}{F}\right) \tag{14.51}$$

使土坡破坏的最小 F 就是土坡的稳定安全系数 F_s。在实际的有限元分析中，所谓破坏是指在指定的最大迭代次数内，分析不收敛。因为，在指定的最大迭次数内，如果算法不能收敛，就意味着没有发现同时既能满足摩尔－库仑破坏准则或 D-P 准则又能满足整体平衡的应力分布，如果算法不能满足这些准则，则说明破坏已经发生了。边坡破坏和数值上的不收敛同时发生，并且伴随着网格中节点位移的显著增加。

1985 年，Tan 和 Donald 提出了一种利用有限元求解得到的节点位移来确定安全系数的简单图解法，称为节点位移法（NDM）。其计算过程和有限元强度折减法相同，不同的是，NDM 通过有限元计算跟踪某一节点的位移 δ 随 F 的变化关系。F 增加时，δ 不断增大，如图 14.7 所示。尖角点给出了 F 的临界值，即为边坡的稳定安全系数。

这种方法包含了如下假定：

1）材料强度参数的降低不影响其他性质；

2）尽管有限元分析是建立在小变形理论基础上的，但在出现尖角点前后，δ 的计算是有

效的。

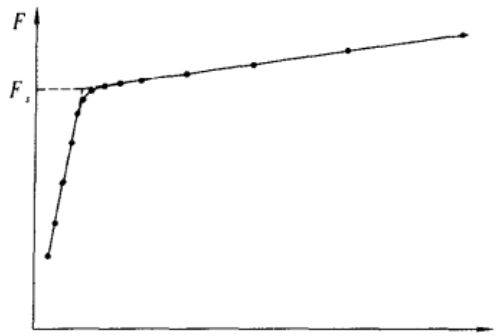


图 14.7 节点位移随 F 的变化图

我们也可以采用边坡内广义剪应变的分布图来判断边坡是否破坏。这一标准是基于在一些模型试验中，研究者发现边坡破坏时形成了明显的剪切破坏带，在边坡内自坡底向上贯通。根据有限元计算结果，绘制边坡内广义剪应变的分布图，如果在某一折减系数下，土体内某一幅值的广义剪应变自坡底部下方向坡顶上方贯通，则认为此前的折减系数为安全系数。该办法图形显示清楚，物理意义明确，但需借助图形的可视化技术。

14.4.3 算例

例 14-2，均质土坡稳定分析算例[28]

此例为一均质土坡。坡高 H=20m，土容重 γ=25kN/m^3，粘聚力 c=42kPa，内摩擦角 ϕ=17°。表 14.6 列出了不同内摩擦角条件下，不同屈服准则得到的稳定安全系数。由表可见，DP4 准则与简化 Bishop 法所得的稳定安全系数最为接近，其误差的平均值为 5.7%，最大误差小于 8%，且离散度也很小。而 DP1 准则的平均误差为 29.5%，采用 DP2 和 DP3 准则所得计算结果的离散度也很大。因此在数值分析中可以用 DP4 准则代替摩尔－库仑准则。

表 14.6 不同屈服准则所得最小安全系数

	ϕ（°）				
	0.1	10	25	35	45
DP1	0.525	1.044	1.769	2.254	3.051
DP2	0.525	0.930	1.332	1.530	1.887
DP3	0.454	0.848	1.279	1.499	1.870
DP4	0.477	0.896	1.396	1.689	2.182
简化 Bishop 法	0.494	0.846	1.316	1.623	2.073
(DP1-Bishop)/Bishop	0.063	0.234	0.344	0.355	0.472
(DP2-Bishop)/Bishop	0.063	0.099	0.012	-0.080	-0.090
(DP3-Bishop)/Bishop	-0.081	0.002	-0.028	-0.099	-0.098
(DP4-Bishop)/Bishop	-0.034	0.059	0.061	0.041	0.053

郑颖等人的研究还发现：

（1）对同一边坡，不论采用关联流动法则还是非关联流动法则，计算结果相差不大。这是因为它们只与坡体的体积变形有关，而在边坡稳定分析中，坡体常常为无约束天然坡体，体积变形对坡体稳定影响并不明显。然而，从破坏时位移大小及塑性区的分布来看，还是会

有一些差异，有时并不能简单的忽略这种差异。

（2）网格疏密对计算精度的影响甚至大于单元类型的影响，精度较低的单元，可通过加密网格来达到较高的精度。对于有限元强度折减法，有限元网格不能太稀，否则结果将不可用。通过大量算例证实，对于 4 节点矩形单元，当单元密度达到第 $10m^2$ 不少于 3 个节点时，计算精度较为理想，如果再增加节点，计算精度还能提高，但此时耗费的机时也将成倍增长。

（3）边界范围的大小在有限元法中对计算结果的影响比在传统极限平衡法中表现的更为敏感。在极限平衡法中只要所求滑动面在边界之内就不会对计算结果有影响，安全系数只与划分的土条有关，而与土条外的区域无关。有限元法则不然，边界的大小直接影响到应力、应变的分布。分析表明，坡脚到侧边的距离对计算结果的影响最不敏感，底部边界次之，坡顶到侧边界的距离影响最大。研究者建议：当坡角到侧边界的距离为坡高的 1.5 倍，坡顶到侧边界的距离为坡高的 2.5 倍，且上下边界总高不低于 2 倍坡高时，计算精度比较理想。

例 14-3　Talbingo 坝

图 14.8 所示高为 162 米的 Talbingo 心墙坝。强度参数从 100%的设计值（$F=1.0$）降到 40%（$F=2.50$）。图 14.9 示一典型节点的位移变化过程，按节点位移法确定的安全的范围大约在 2.05～2.14 之间。对应于 $F=2.06$ 和 2.11 的位移矢量参见图 14.10。在 $F=2.06$ 时，无明显大变形出现，坝中无明显滑动面。在 $F=2.11$ 时，在坝的上游坡堆石料区、过渡料区和心墙料区可以很明显看到大变形出现。由图 14.10 可以看到一明显的滑动面。用条分法计算所得通过心墙的滑动面的安全系数为 2.06。说明两者是一致的。

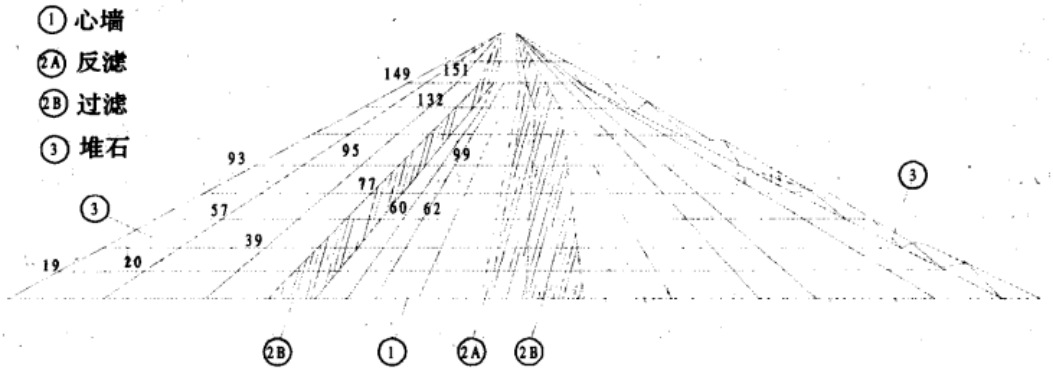


图 14.8　Talbingo 心墙坝

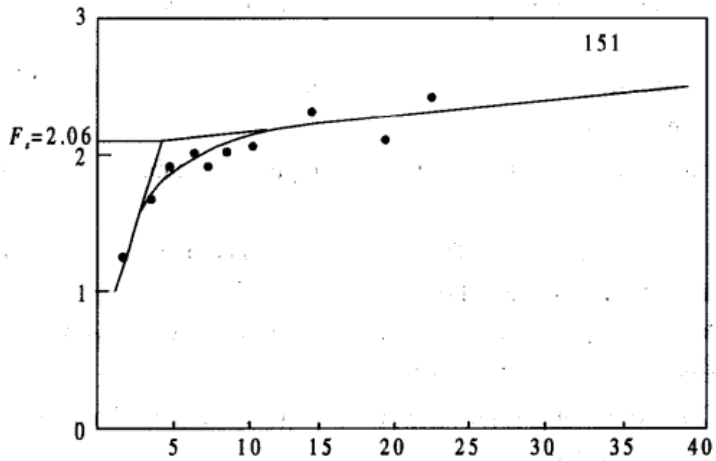


图 14.9　节点位移变化过程

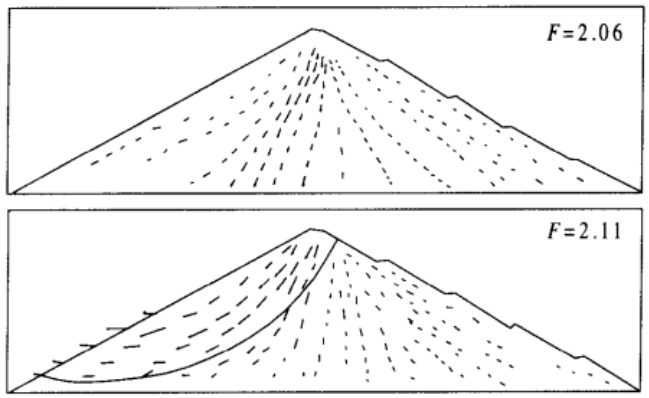


图 14.10 不同折减系数下的位移矢量图

参考文献

[1] 王恭先.面向 21 世纪我国滑坡灾害防治的思考.见:兰州滑坡泥石流学术研讨会文集.兰州:兰州大学出版社，1998

[2] 赵明阶，何光春，王多垠.边坡工程处治技术. 北京:人民交通出版社，2003

[3] A.W. Bishop.The use of the slip circle in the stability analysis of slopes.*Geotechnique*，1955，5(1)

[4] 钱家欢，殷宗泽.土工原理与计算. 北京:中国水利水电出版社，1996

[5] 陈祖煜著.土质边坡稳定分析——原理、方法、程序. 北京:中国水利水电出版社，2003

[6] D.G. Fredlund. Analytical methods for slope stability analysis. Proc. of the Fourth Int. Symposum on Landslides，1984

[7] J. M. Duncan. State of the art: Limit equilibrium and finite element analysis of slopes.*Journal of Geotech. Engineering, ASCE*，1996，122(7)

[8] 王成华，夏绪勇.边坡稳定分析中的临界滑动面搜索方法述评.四川建筑科学研究，2002，28(3)

[9] R. Baker， M. Garber. Variation approach to slope stability. Proc. 9th Int. Conf. on Soil Mech. and Found. Engng., 1977

[10]E. Castillo，J. Revilla.The calculus of variations and the stability of slopes.Proc. 9th Int. Conf. on Soil Mech. and Found. Engng., 1977，2

[11]J.Revilla, E.Castillo, The calculus of variations applied to stability of slops, *Geotechnique*，1977，27(1)

[12]T. Ramamurthy，C.G.P. Narayan，V.P. Bhatkar. Variational methods for slope stability analysis. Proc. 9th Int. Conf. on Soil Mech. and Found. Engng., 1977

[13] De Josselin，De Jong. Application of the calculus of variation to vertical cut off in cohesive frictionless soil. *Geotechnique*，1980，30(1)

[14]A. Luceno，E. Castillo. Evaluation of variational methods in slope analysis. Proc. 1st Int. Symp. on Landslides, Sarita Prakashan, New Delhi，1980

[15]T.B. Celestino，J.M. Duncan. Simplified search for non-circular slip surface. Proc. 10th Int. Conf.

on Soil Mech. and Found. Engrg.，A.A. Balkema，Rotterdam，the Netherlands，1981

[16]K. Arai，K. Tagyo. Determination of noncircular slip surface giving the minimum factor of safety in slope stability analysis. *Soils and Foundations*，1985，25(1)

[17]K.S. Li，W. White. Rapid evaluation of the critical surface in slope stability problems[J]. Int. J. for Numer. and Anal. Meth. In Geomech.，1987，11(5):449-473

[18]孙君实. 条分法的数值分析. 岩土工程学报，1984，6(2)

[19]T. Yamagami, Y.Ueta. Search for noncircular slip surfaces by the Mrgenstern-Preice method. Proc. of 6th Int. Conf. on Numerical Methods in Geomechanics，Innsbruck，1988.

[20]陈祖煜，邵长明. 最优化方法在确定边坡最小安全系数方面的应用. 岩土工程学报，1988，（4）

[21]J.Z. Zhou, D.J. Williams, W.L. Xiong, Search for critical slip surfaces based on finite element method, *Can. Geotech. J.*, 1995, 32

[22]史恒通，王成华. 土坡有限元稳定分析若干问题探讨. 岩土力学，2000，21（2）

[23]金小惠. 饱和土坡渗流-变形-稳定性的非线性耦合有限元分析. 天津大学学报，2002

[24]O.C. Zienkiewicz, C.Humpheson, R.W. Lewis, Associated and non-associated visco-plasticity and plasticity in soil mechanics. *Geotechnique*, 1975, 25(4)

[25]宋二祥. 土工结构安全系数的有限元计算. 岩土工程学报，1997，19（2）

[26]D.V. Griffiths，P.A. Lane. Slope stability analysis by finite elements. *Geotechnique*, 1999, 49(3)

[27]郑颖人，赵尚毅，张鲁渝. 用有限元强度折减法进行边坡稳定分析. 中国工程科学，2002，4（10）

[28]张鲁渝，郑颖人，赵尚毅，时卫民.有限元强度折减系数法计算土坡稳定安全系数的精度研究.水利学报，2003，（1）

VFEAP 主要源文件文本

为方便读者阅读，以下列出 VFEAP 程序中主要源文件的文本清单。

附 1　VFEAP.f90 文本

```
!  ======================================================================
!                          通用结构分析程序——VFEAP v3.0
!              程序以中文 Windows3.1 和 Fortran90(Fortran Power Station V4)
!          为平台，采用波阵解法求解方程，采用波阵逆迭代法求解特征值问题，求
!          解规模受内外存限制。可调数组的使用使得内存空间得到充分利用，动态
!          连接库的使用使得程序模块化，宏命令有效地组织各种有限元算法. 浙江
!          大学岩土工程研究所凌道盛对本程序拥有所有权。版权所有，不得复制。
!  ======================================================================
        INTEGER * 4 FUNCTION WinMain( hInstance, hPrevInstance,          &
                                      lpszCmdLine, nCmdShow )
        !DEC$ IF DEFINED(_X86_)
        !DEC$ ATTRIBUTES STDCALL, ALIAS : '_WinMain@16' :: WinMain
        !DEC$ ELSE
        !DEC$ ATTRIBUTES STDCALL, ALIAS : 'WinMain' :: WinMain
        !DEC$ END IF

        USE User32
        USE kernel32
        USE dflogm
        USE dfcom
        USE dfauto
        USE VFEAPGlobals
        IMPLICIT NONE

        INTEGER * 4 hInstance
        INTEGER * 4 hPrevInstance
        INTEGER * 4 lpszCmdLine
        INTEGER * 4 nCmdShow
        INCLUDE 'resource.fd'
        TYPE (T_MSG)                mesg
```

```
INTEGER * 4                    ret
LOGICAL * 4                    lret

gpSetRunFlag = 0
ghInstance = hInstance
ghModule = GetModuleHandle(NULL)
ghInstVFEAPCom = LoadLibrary( "VFEAPCom.dll" )
ret = GetLastError()

ghWndMain = NULL
CALL COMINITIALIZE(ret)
lret = DlgInit(IDD_VFEAP_DIALOG, gdlg)

IF( lret == .TRUE. ) THEN
  lret = DlgModeless( gdlg, nCmdShow )
  IF( lret == .TRUE. ) THEN
    ghWndMain = gDlg%hWnd
    CALL ShowMessage( '欢迎使用 VFEAP...' )
    IF( gpSetRunFlag .EQ. 0 .AND. ghInstVFEAPCom .NE. 0 ) THEN
      gpSetRunFlag = GetProcAddress( ghInstVFEAPCom, "SetRunFlag"C )
    END IF
    IF( gpSetRunFlag .NE. 0 ) CALL SetRunFlag( 1 )
    CALL VFEAP
    IF( gpSetRunFlag .NE. 0 ) CALL SetRunFlag( 0 )
    IF( ghInstVFEAPCom .NE. 0 ) THEN
      lRet = FreeLibrary( ghInstVFEAPCom )
    END IF
    CALL PostQuitMessage( 0 )
    DO WHILE( GetMessage( mesg, NULL, 0, 0 ) )
      IF( DlgIsDlgMessage(mesg) .EQV. .FALSE. ) THEN
        lret = TranslateMessage( mesg )
        ret  = DispatchMessage( mesg )
      END IF
    END DO
    CALL DlgUninit(gdlg)
    CALL COMUNINITIALIZE()
    WinMain = mesg.wParam
    RETURN
  END IF
END IF
ret = MessageBox( ghwndMain, "VFEAP 初始化错误!"C, "错误"C, MB_OK )
CALL COMUNINITIALIZE()
WinMain = 0
END

SUBROUTINE VFEAP
USE CtrlData
USE MeshData
```

```
      USE CentData
      USE LoadData
      USE FrontData
      USE ResolveData
      USE AutoMeshData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )

      nerrc = 0
      CALL RunTime( 1 )
      CALL OpenFile
      CALL InitCtrl
      CALL InitMesh
      CALL InitCent
      CALL InitLoad
      CALL InitFront
      CALL InitAutoMesh
      CALL InitReSolve
      CALL InitMemory

      CALL Input
      CALL Execute
      CALL ErrorMsg
      CALL RunTime( 2 )
      CALL CloseFile

      CALL DelMesh
      CALL DelCent
      CALL DelLoad
      CALL DelFront
      CALL DelReSolve
      CALL DelAutoMesh
      CALL DelCtrl
      RETURN
      END

      SUBROUTINE OpenFile
!.....
!     模块功能
!          打开输入,输出文件,读入控制信息.
!.....
      USE CtrlData
      USE FileData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
!.....
      CALL FileName
      OPEN( 11, FILE = sdatf, STATUS = 'OLD' )
      OPEN( 12, FILE = soutf, STATUS = 'UNKNOWN' )
      OPEN( 13, FILE = smacf, STATUS = 'OLD' )
```

```
      OPEN( 14, FILE = sinif, STATUS = 'UNKNOWN' )
      OPEN( 15, FILE = smshf, STATUS = 'UNKNOWN' )
      OPEN( 16, FILE = sdisf, STATUS = 'UNKNOWN' )
      OPEN( 17, FILE = sstrf, STATUS = 'UNKNOWN' )
      OPEN( 18, FILE = ssavf, FORM   = 'UNFORMATTED' )
      OPEN( 19, FILE = sdsvf, FORM   = 'UNFORMATTED' )
      OPEN( 20, FILE = sstaf, STATUS = 'UNKNOWN' )
      OPEN( 21, FILE = shisf, STATUS = 'UNKNOWN' )
      OPEN( 22, FILE = spref, STATUS = 'UNKNOWN' )
      OPEN( 23, FILE = strnf, STATUS = 'UNKNOWN' )
      CALL ShowMessage( '读入网格数据...' )
      CALL ShowMessage( '读控制信息...' )
      WRITE( 12, 2000 )
      READ(  11, 2200 ) title
      WRITE( 12, 2400 ) title
      CALL ShowTitle( title )
      READ(  11, 2600 ) ntypc, npoin, nelem, ndime,                    &
                        mdofn, nmats, nload, nslnt
      READ(  11, 2600 ) mnode, nnodp, nnodg, nprop,                    &
                        mnodf, medgf, mplnf, nprnc
      READ(  11, 2600 ) nmssp, nstfp, ndmpp, nmode,                    &
                        nlnkc, nfmtc
      IF( ntypc .LE. 6 .AND. ntypc .GE. 0 ) THEN
        WRITE( 15, 2600 ) ntypc
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 0 ) WRITE( 12, 2800 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 1 ) WRITE( 12, 2801 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 2 ) WRITE( 12, 2802 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 3 ) WRITE( 12, 2803 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 4 ) WRITE( 12, 2804 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 5 ) WRITE( 12, 2805 )
        IF( nprnc .NE. 0 .AND. ntypc .EQ. 6 ) WRITE( 12, 2806 )
      ELSE
        nerrc = 1
        RETURN
      END IF
      IF( nprnc .NE. 0 ) THEN
        WRITE( 12, 3200 ) npoin, nelem, ndime, mdofn
        WRITE( 12, 3400 ) nmats, nload, mnode, nslnt
        WRITE( 12, 3600 ) nnodp, nnodg, nprop, mnodf
        WRITE( 12, 3800 ) medgf, mplnf, nmssp, nstfp
        WRITE( 12, 4000 ) ndmpp, nmode
        IF( nlnkc .EQ. 0 ) WRITE( 12, 4200 )
        IF( nlnkc .NE. 0 ) WRITE( 12, 4201 )
      END IF
      RETURN
200   nerrc = 5235
      RETURN
2000  FORMAT( // 30x, '通用结构分析程序'// )
```

```
2200    FORMAT( a70    )
2400    FORMAT( //2x, 70( '=' ) // 2x, a70 // 2x, 70( '=' ) // )
2600    FORMAT( 10I5 )
2800    FORMAT( 2x, '问题类型   : 静力分析' )
2801    FORMAT( 2x, '问题类型   : 动特性分析' )
2802    FORMAT( 2x, '问题类型   : 动响应分析' )
2803    FORMAT( 2x, '问题类型   : 稳定性分析' )
2804    FORMAT( 2x, '问题类型   : 阻尼动特性分析' )
2805    FORMAT( 2x, '问题类型   : 阻尼动响应分析' )
2806    FORMAT( 2x, '问题类型   : 阻尼动稳定性分析' )
3200    FORMAT( 2x, '节点总数   : ', I5, 10x, '单元总数:     ', I5 /                &
              2x, '问题维数   : ', I5, 10x, '节点自由度: ', I5 )
3400    FORMAT( 2x, '材料总数   : ', I5, 10x, '载荷工况数: ', I5 /                 &
              2x, '单元节点数: ', I5, 10x, '斜自由度数: ', I5 )
3600    FORMAT( 2x, '面载节点数: ', I5, 10x, '线载节点数: ', I5 /                  &
              2x, '材性参数数: ', I5, 10x, '集中载荷数: ', i5 )
3800    FORMAT( 2x, '线布载荷数: ', I5, 10x, '面布载荷数: ', i5 /                  &
              2x, '集中质量数: ', I5, 10x, '集中刚度数: ', I5 )
4000    FORMAT( 2x, '集中阻尼数: ', I5, 10x, '最大模态数: ', I5 )
4200    FORMAT(/2x, '注意: @坐标相同或相近的节点将不被自动连接@' / )
4201    FORMAT(/2x, '注意: @坐标相同或相近的节点将被自动连接@' / )
        RETURN
        END

        SUBROUTINE CloseFile
        WRITE( 16, 2000 ) 0
        WRITE( 17, 2000 ) 0
        WRITE( 20, 2000 ) 0
        WRITE( 23, 2000 ) 0
        CLOSE( 11 )
        CLOSE( 12 )
        CLOSE( 13 )
        CLOSE( 14 )
        CLOSE( 15 )
        CLOSE( 16 )
        CLOSE( 17 )
        CLOSE( 18 )
        CLOSE( 19 )
        CLOSE( 20 )
        CLOSE( 21 )
        CLOSE( 22 )
        CLOSE( 23 )
2000    FORMAT( I5 )
        RETURN
        END
```

附 2　Input.f90 文本

```
      SUBROUTINE Input
      USE CtrlData
      IF( nfmtc .EQ. 0 ) CALL FixInput
      IF( nfmtc .NE. 0 ) CALL MacroInput
      RETURN
      END

      SUBROUTINE FixInput
      CALL RdCrd
      CALL RdLnk
      CALL RdElm
      CALL RdFix
      CALL RdMat
      CALL RdFrc
      CALL RdCnt
      CALL RdSlnt
      CALL RdOptAxis
      RETURN
      END

      SUBROUTINE MacroInput
      USE CtrlData
      USE MeshData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      CHARACTER strng * 80, mwork * 10
      IF( nerrc .GT. 0 ) RETURN

      label = 1
      ifcrd = 0
      ifelm = 0
      iffix = 0
      ifmat = 0
      iffrc = 0
      ifcnt = 0
      ifsln = 0
      iflnk = 0
      ifaut = 0
      ifmid = 0

!.....节点连接信息初始化.
      DO ipoin = 1, npoin
        llnks( ipoin ) = ipoin
      END DO
```

```
      DO WHILE( label .EQ. 1 )
        READ( 11, '( a80 )', END = 400 ) strng
        CALL ReLocate( strng, nerrc )
        IF( nerrc .GT. 0 ) RETURN
        mwork = strng( 1:10 )
        IF( mwork .EQ. 'end         ' ) THEN
          RETURN
        ELSE IF( mwork .EQ. 'coordinate' ) THEN
          CALL RdCrd
          ifcrd = ifcrd + 1
        ELSE IF( mwork .EQ. 'element     ' ) THEN
          CALL RdElm
          ifelm = ifelm + 1
        ELSE IF( mwork .EQ. 'fix         ' ) THEN
          CALL RdFix
          iffix = iffix + 1
        ELSE IF( mwork .EQ. 'material    ' ) THEN
          CALL RdMat
          ifmat = ifmat + 1
        ELSE IF( mwork .EQ. 'force       ' ) THEN
          CALL RdFrc
          iffrc = iffrc + 1
        ELSE IF( mwork .EQ. 'link        ' ) THEN
          CALL RdLnk
          iflnk = iflnk + 1
        ELSE IF( mwork .EQ. 'center      ' ) THEN
          CALL RdCnt
          ifcnt = ifcnt + 1
        ELSE IF( mwork .EQ. 'slent       ' ) THEN
      CALL RdSlnt
          ifsln = ifsln + 1
        ELSE IF( mwork .EQ. 'autolink    ' ) THEN
          CALL AutoLink( strng )
          ifaut = ifaut + 1
        ELSE IF( mwork .EQ. 'midnode     ' ) THEN
          CALL MidNode
          ifmid = ifmid + 1
        END IF
        IF( nerrc .NE. 0 ) RETURN
      END DO
      RETURN
400   WRITE( 12, 2000 )
      nerrc = 3004
      RETURN
2000  FORMAT( ' ' )
      END

      SUBROUTINE RdCrd
```

```
!.....
!       模块功能
!           读写节点坐标.
!.....
        USE CtrlData
        USE MeshData
        USE AutoMeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        DIMENSION corpi( 3 ), corpj( 3 ), cstep( 3 )
        IF( nerrc .GT. 0 ) RETURN
        pai = datan( 1.0D0 ) / 45.0D0
        CALL ShowMessage( '读节点坐标...' )

        jgene = 0
        iflag = 1
        DO WHILE( iflag .EQ. 1 )
         READ( 11, 2000 ) ipoin, ictyp, igene, corpi, tempi, vdimi
         IF( ipoin .EQ. 0 .OR. ipoin .GT. npoin ) THEN
            iflag = 0
         ELSE
          llnks( ipoin ) = llnks( ipoin ) + 1
          temps( ipoin ) = tempi
          vdims( ipoin ) = vdimi
          IF( ictyp .EQ. 0 ) THEN
           DO idime = 1, ndime
            coord( idime, ipoin ) = corpi( idime )
           END DO
          ELSE IF( ictyp .EQ. 1 ) THEN
           theta = corpi( 2 ) * pai
           coord( 1, ipoin ) = corpi( 1 ) * dcos( theta )
           coord( 2, ipoin ) = corpi( 1 ) * dsin( theta )
           IF( ndime .EQ. 3 ) coord( 3, ipoin ) = corpi( 3 )
          ELSE IF( ictyp .EQ. 2 ) THEN
           alpha = corpi( 2 ) * pai
           theta = corpi( 3 ) * pai
           coord( 1, ipoin ) = corpi(1) * dsin( alpha ) * dcos( theta )
           coord( 2, ipoin ) = corpi(1) * dsin( alpha ) * dsin( theta )
           coord( 3, ipoin ) = corpi(1) * dcos( alpha )
          ELSE
           WRITE( 12, 2200 ) ictyp, ipoin
           nerrc = 1001
           RETURN
          END IF
          IF( jgene .NE. 0 ) THEN
           IF( ictyp .NE. jctyp ) THEN
            WRITE( 12, 2400 ) jpoin, ipoin
            nerrc = 1002
            RETURN
```

```
      END IF
      nsegm = ( ipoin - jpoin ) / jgene
      IF( nsegm * jgene .NE. ipoin - jpoin ) THEN
       WRITE( 12, 2600 ) jgene, jpoin, ipoin
       nerrc = 1003
       RETURN
      END IF
      IF( nsegm .NE. 0 ) THEN
       tmpst = ( tempi - tempj ) / nsegm
       dimst = ( vdimi - vdimj ) / nsegm
       DO idime = 1, ndime
        cstep( idime ) = (corpi( idime ) - corpj( idime )) / nsegm
       END DO
       DO kpoin = jpoin + jgene, ipoin - jgene, jgene
        llnks( kpoin ) = llnks( kpoin ) + 1
        tempj = tempj + tmpst
        vdimj = vdimj + dimst
        DO idime = 1, ndime
         corpj( idime ) = corpj( idime ) + cstep( idime )
        END DO
        temps( kpoin ) = tempj
        vdims( kpoin ) = vdimj
        IF( jctyp .EQ. 0 ) THEN
         DO idime = 1, ndime
          coord( idime, kpoin ) = corpj( idime )
         END DO
        ELSE IF( jctyp .EQ. 1 ) THEN
         theta = corpj( 2 ) * pai
         coord( 1, kpoin ) = corpj( 1 ) * dcos( theta )
         coord( 2, kpoin ) = corpj( 1 ) * dsin( theta )
         IF( ndime .EQ. 3 ) coord( 3, kpoin ) = corpj( 3 )
        ELSE IF( jctyp .EQ. 2 ) THEN
         alpha = corpj( 2 ) * pai
         theta = corpj( 3 ) * pai
         coord( 1, kpoin ) = corpj( 1 ) * dsin( alpha ) *                    &
                                                  dcos( theta )
         coord( 2, kpoin ) = corpj( 1 ) * dsin( alpha ) *                    &
                                                  dsin( theta )
         coord( 3, kpoin ) = corpj( 1 ) * dcos( alpha )
        END IF
       END DO
      END IF
     END IF
     jpoin = ipoin
     jgene = igene
     jctyp = ictyp
     tempj = tempi
     vdimj = vdimi
```

```
          corpj( 1 ) = corpi( 1 )
          corpj( 2 ) = corpi( 2 )
          corpj( 3 ) = corpi( 3 )
         END IF
        END DO

        DO ipoin = 1, npoin
         IF( llnks( ipoin ) .EQ. 0 ) THEN
           WRITE( 12, 2800 ) ipoin
         ELSE IF( llnks( ipoin ) .GT. 1 ) THEN
           WRITE( 12, 3000 ) ipoin
         END IF
         llnks( ipoin ) = 0
        END DO

        IF( nprnc .NE. 0 ) THEN
         WRITE( 12, 3200 )
          corpi( 1 ) = 0.0D0
          corpi( 2 ) = 0.0D0
          corpi( 3 ) = 0.0D0
         DO ipoin = 1, npoin
          DO idime = 1, ndime
           corpi( idime ) = coord( idime, ipoin )
          END DO
          tempi = temps( ipoin )
          vdimi = vdims( ipoin )
          WRITE( 12, 3400 ) ipoin, corpi, tempi, vdimi
         END DO
        END IF
2000    FORMAT( 3i5, 5f15.5 )
2200    FORMAT( 2x, '错误: 无效的坐标类型', i2, ',  节点', i5 )
2400    FORMAT( 2x, '错误: 自动坐标不适用于不同坐标类型, ',                    &
                    '节点', i5, '和', i5 )
2600    FORMAT( 2x, '错误: 无效的自动步长', I5, '节点', i5, '和', i5 )
2800    FORMAT( 2x, '警告:  节点', I5, '  没有定义' )
3000    FORMAT( 2x, '警告:  节点', I5, '  重复定义' )
3200    FORMAT( //10x, '= = = =  节点坐标  = = = ='                          &
              // 3x, '节点       x-坐标          y-坐标          z-坐标'      &
                     '          节点温度        单元尺度' )
3400    FORMAT(     2X, I5, 5F13.5 )
        RETURN
        END

        SUBROUTINE RdLnk
!.....
!       模块功能
!           读取节点连接信息.
```

```
!.....
      USE CtrlData
      USE MeshData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION lwork( 10 )
      IF( nerrc .GT. 0 ) RETURN

!.....节点连接信息初始化.
      DO ipoin = 1, npoin
        llnks( ipoin ) = ipoin
      END DO
!.....坐标相同的节点自动连接.
      IF( nlnkc .NE. 0 ) THEN
        DO ipoin = 1, npoin
          IF( ipoin .EQ. llnks( ipoin ) ) THEN
            DO jpoin = ipoin + 1, npoin
              IF( jpoin .EQ. llnks( jpoin ) ) THEN
                displ = DABS( coord(1, ipoin) - coord(1, jpoin) )
                IF( displ .LT. 1.5D-5 ) THEN
                  nflag = 0
                  DO idime = 2, ndime
                    displ = DABS( coord( idime, ipoin ) -                    &
                                  coord( idime, jpoin ) )
                    IF( displ .GT. 1.5D-5 ) nflag = 1
                  END DO
                  IF( nflag .EQ. 0 ) llnks( jpoin ) = ipoin
                END IF
              END IF
            END DO
          END IF
        END DO
      END IF
!.....读入节点连接信息
      nflag = 1
      DO WHILE( nflag .EQ. 1 )
        kpoin = 0
        ipoin = npoin
        READ( 11, 2000, ERR = 100 ) lwork
        DO inode = 1, 10
          jpoin = lwork( inode )
          IF( jpoin .NE. 0 ) THEN
            IF( jpoin .GT. npoin .OR. jpoin .LT. 0 ) THEN
              WRITE( 12, 2200 ) jpoin
              nerrc = 24356
              RETURN
            ELSE
              ipoin = jpoin
              kpoin = kpoin + 1
```

```
            END IF
          END IF
        END DO
        IF( kpoin .EQ. 0 ) THEN
          nflag = 0
        ELSE IF( kpoin .EQ. 1 ) THEN
          WRITE( 12, 2400 )
          nerrc = 24356
          RETURN
        ELSE
          DO inode = 1, 10
            jpoin = lwork( inode )
            IF( jpoin .GT. 0 .AND. jpoin .LE. npoin ) THEN
              DO WHILE( ipoin .NE. llnks( ipoin ) )
                ipoin = llnks( ipoin )
              END DO
              DO WHILE( jpoin .NE. llnks( jpoin ) )
                jpoin = llnks( jpoin )
              END DO
              IF( ipoin .LT. jpoin ) llnks( jpoin ) = ipoin
              IF( ipoin .GT. jpoin ) llnks( ipoin ) = jpoin
            END IF
          END DO
        END IF
      END DO
      IF( nprnc .NE. 0 ) THEN
        WRITE( 12, 2600 )
        DO ipoin = 1, npoin
          WRITE( 12, 2800 ) ipoin, llnks( ipoin )
        END DO
      END IF
      RETURN
100   WRITE( 12, 3000 )
      nerrc = 59759
      RETURN
2000  FORMAT( 10I5 )
2200  FORMAT( 2x, '错误: 无效节点 ', I5 )
2400  FORMAT( 2x, '错误: 至少有两个节点才能连接在一起.' )
2600  FORMAT( ///2x, 60('=') // 20x, '节点连接信息'                    &
              //2x, 60('=') /// )
2800  FORMAT( 16x, I5, 10x, I5 )
3000  FORMAT( //2x, '错误: 读节点连接信息错.' )
      END

      SUBROUTINE RdElm
!.....
!     模块功能
```

```
!              读写单元信息.
!.....
      USE CtrlData
      USE MeshData
      USE AutoMeshData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION lnodt( 10 ), ldimt( 3 ), lwork( 100 )

      IF( nerrc .GT. 0 ) RETURN
      CALL ShowMessage( '读单元定义...' )

      jelem = 0
      jgene = 0
      iflag = 1
      DO WHILE( iflag .EQ. 1 )
       READ( 11, 2000 ) ielem, imats, igene, lnodt, ldimt
       IF( ielem .EQ. 0 .OR. ielem .GT. nelem ) THEN
         iflag = 0
       ELSE
        IF( imats .LT. 1 .OR. imats .GT. nmats ) THEN
         WRITE( 12, 2200 ) imats, ielem
         nerrc = 1004
         RETURN
        END IF
        lmats( ielem ) = imats
        DO inode = 1, MIN0( 10, mnode )
         lnods( inode, ielem ) = lnodt( inode )
        END DO
        DO iloca = 11, mnode, 10
         READ( 11, 2000 ) izero, izero, izero, lnodt
         DO inode = iloca, MIN0( iloca + 9, mnode )
          lnods( inode, ielem ) = lnodt( inode - iloca + 1 )
         END DO
        END DO
        DO idime = 1, ndime
         ldims( idime, ielem ) = ldimt( idime )
        END DO
        IF( jgene .NE. 0 ) THEN
         IF( imats .NE. jmats ) THEN
          WRITE( 12, 2400 ) jelem, ielem
          nerrc = 1005
          RETURN
         END IF
         nsegm = ( ielem - jelem ) / jgene
         IF( nsegm * jgene .NE. ielem - jelem ) THEN
          WRITE( 12, 2600 ) jgene, jelem, ielem
          nerrc = 1006
          RETURN
```

```
      END IF

      DO inode = 1, mnode
       lwork( inode ) = ( lnods( inode, ielem ) -                    &
                          lnods( inode, jelem ) ) / nsegm
       IF( lwork( inode ) * nsegm .NE. lnods( inode, ielem ) -       &
           lnods( inode, jelem ) ) THEN
        WRITE( 12, 2800 ) inode, jelem, ielem
        nerrc = 1007
        RETURN
       END IF
      END DO

      DO idime = 1, ndime
       ldimt( idime ) = ( ldims( idime, ielem ) -                    &
                          ldims( idime, jelem ) ) / nsegm
       IF( ldimt( idime ) * nsegm .NE. ldims( idime, ielem ) -       &
           ldims( idime, jelem ) ) THEN
        WRITE( 12, 2900 ) idime, jelem, ielem
        nerrc = 1007
        RETURN
       END IF
      END DO

      DO kelem = jelem + jgene, ielem - jgene
       lmats( kelem ) = jmats
       DO inode = 1, mnode
        lnods( inode, kelem ) = lnods( inode, kelem - jgene ) +      &
        lwork( inode )
       END DO
       DO idime = 1, ndime
        ldims( idime, kelem ) = ldims( idime, kelem - jgene ) +      &
        ldimt( idime )
       END DO
      END DO
     END IF
     jelem = ielem
     jmats = imats
     jgene = igene
    END IF
   END DO

   DO ielem = 1, nelem
     IF( lmats( ielem ) .EQ. 0 ) WRITE( 12, 3000 ) ielem
   END DO
   IF( nprnc .NE. 0 ) THEN
    WRITE( 12, 3200 )
    DO ielem = 1, nelem
```

```
          iwrot = 0
          imats = lmats( ielem )
          DO idime = 1, ndime
           ldimt( idime ) = ldims( idime, ielem )
          END DO
          DO inode = 1, mnode
           iloca = MOD( inode, 10 )
           IF( iloca .EQ. 0 ) iloca = 10
           lnodt( iloca ) = lnods( inode, ielem )
           IF( inode .EQ. mnode ) THEN
            DO jloca = iloca + 1, 10
             lnodt( jloca ) = 0
            END DO
           END IF
           IF( iloca .EQ. 10 .OR. inode .EQ. mnode ) THEN
            iwrot = iwrot + 1
            IF( iwrot .EQ. 1 ) THEN
             WRITE( 12, 3400 ) ielem, imats, lnodt, ldimt
            ELSE
             WRITE( 12, 3600 ) lnodt
            END IF
           END IF
          END DO
         END DO
        END IF
        RETURN
2000    FORMAT( 16I5 )
2200    FORMAT( 2x, '错误: 无效的材料号', i2, ', 单元', i5 )
2400    FORMAT( 2x, '错误: 不同类型单元', I5, '和',                          &
              I5, '间不可自动生成' )
2600    FORMAT( 2x, '错误: 无效的单元步长', i5,                              &
                  ', 单元', i5, '和', i5 )
2800    FORMAT( 2x, '错误: 无效的自动步长, 节点', i5,                        &
                  ',  单元', i5, '和', i5 )
2900    FORMAT( 2x, '错误: 无效的自动步长, 维数', i5,                        &
                  ',  单元', i5, '和', i5 )
3000    FORMAT( 2x, '警告: 单元', I5, ' 没有定义' )
3200    FORMAT( // 18x, ' ==== 单 元 信 息 ==== ' //                         &
                 2x, ' 单元 材料   节点  节点  节点  节点  节点',            &
                   '  节点  节点  节点  节点  节点'                  &
                   '  分段  分段  分段' )
3400    FORMAT(     2x, 2I5, 1x, 13I6 )
3600    FORMAT(    13x, 10I6 )
        END

        SUBROUTINE RdFix
```

```
!.....
!       模块功能
!           读写位移约束信息.
!.....
        USE CtrlData
        USE MeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )

        IF( nerrc .GT. 0 ) RETURN
        CALL ShowMessage( '读约束信息...' )

        DO ipoin = 1, npoin
          DO idofn = 1, mdofn
            lfixs( idofn, ipoin ) = 0
            vfixs( idofn, ipoin ) = 0.0D0
          END DO
        END DO
        label = 1
        DO WHILE( label .EQ. 1 )
          READ( 11, 2000, ERR = 200 ) ipoin, idofn, itype, vcord, value
          IF( ipoin .EQ. 0 ) THEN
            label = 0
          ELSE
            IF( idofn .LE.        0 ) nerrc = 1009
            IF( ipoin .LT.-ndime ) nerrc = 1008
            IF( ipoin .GT. npoin ) nerrc = 1008
            IF( idofn .GT. mdofn ) nerrc = 1009
            IF( nerrc .EQ.   1008 ) WRITE( 12, 2200 ) ipoin
            IF( nerrc .EQ.   1009 ) WRITE( 12, 2400 ) idofn, ipoin
            IF( nerrc .GT. 0 ) RETURN
            IF( ipoin .GT. 0 ) THEN
              ipoin = llnks( ipoin )
              lfixs( idofn, ipoin ) = itype + 1
              vfixs( idofn, ipoin ) = value
            ELSE
              idime =-ipoin
              DO ipoin = 1, npoin
                vdisp = DABS( coord( idime, ipoin ) - vcord )
                IF( vdisp .LT. 1.0D-8 ) THEN
                  lfixs( idofn, llnks( ipoin ) ) = itype + 1
                  vfixs( idofn, llnks( ipoin ) ) = value
                END IF
              END DO
            END IF
          END IF
        END DO
        IF( nprnc .NE. 0 ) THEN
          WRITE( 12, 2600 )
```

```
          DO ipoin = 1, npoin
            DO idofn = 1, mdofn
              IF( lfixs( idofn, ipoin ) .NE. 0 ) THEN
                value = vfixs( idofn, ipoin )
                itype = lfixs( idofn, ipoin ) - 1
                WRITE( 12, 2800 ) ipoin, idofn, itype, value
              END IF
            END DO
          END DO
        END IF
        RETURN
200     WRITE( 12, 3000 )
        nerrc = 1010
        RETURN
2000    FORMAT( 3I5, 2F15.5 )
2200    FORMAT( 2x, '错误: 无效的节点号', i5 )
2400    FORMAT( 2x, '错误: 无效的自由度', I3, ', 节点', I5 )
2600    FORMAT( //8x, '==== 位 移 约 束 信 息 ====' //                    &
                  5x, '    节点      自由度            给定位移 ' )
2800    FORMAT(     7x, i5, 6x, I2, 6x, I2, 8x, F15.5 )
3000    FORMAT( 2x, '错误: 读约束代码出错' )
        END

        SUBROUTINE RdMat
!.......
!       模块功能
!           读写材料信息.
!.......
        USE CtrlData
        USE MeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        ALLOCATABLE lprpt(:)

        ALLOCATE( lprpt( mdofn + 3 ), STAT = ierro )
        IF( ierro .NE. 0 ) nerrc = 4
        IF( nerrc .GT. 0 ) RETURN

        CALL ShowMessage( '读材料信息...' )
        IF( nprnc .NE. 0 ) WRITE( 12, 2000 )

        iswth = 0
        nhstr = 0
        nstrh = 0
        nmatp = mdofn + 3
        DO imats = 1, nmats
          READ( 11, 2200 ) jmats, lprpt
          IF( jmats .LT. 1 .OR. jmats .GT. nmats ) nerrc = 1011
```

```
      IF( nprnc .NE. 0 ) THEN
        WRITE( 12, 2400 ) jmats, lprpt( 1 ), lprpt( 2 )
        WRITE( 12, 2600 ) ( lprpt( imatp + 2 ), imatp = 1, mdofn )
      END IF
      IF( nerrc .GT. 0 ) DEALLOCATE( lprpt )
      IF( nerrc .GT. 0 ) RETURN
      ietyp = lprpt(         1 )
      ipara = lprpt( nmatp )
      jetyp = lprps(         1, jmats )
      jpara = lprps( nmatp, jmats )

      IF( jetyp .NE. 0 ) THEN
        IF( ietyp .NE. jetyp .OR. ipara .NE. jpara ) THEN
          WRITE( 12, 2800 ) jmats, jetyp, jpara, ietyp, ipara
          DEALLOCATE( lprpt )
          nerrc = 3401
          RETURN
        END IF
      END IF

      nchck = 0
      DO idofn = 1, mdofn
        nchck = nchck + lprpt( idofn + 2 )
      END DO
      IF( nchck .EQ. 0 ) THEN
        DO idofn = 1, mdofn
          lprpt( idofn + 2 ) = idofn
        END DO
      END IF
      DO imatp = 1, nmatp
        lprps( imatp, jmats ) = lprpt( imatp )
      END DO
      CALL ElmOpt( jmats, iswth )
      IF( nerrc .GT. 0 ) DEALLOCATE( lprpt )
      IF( nerrc .GT. 0 ) RETURN
    END DO
    DEALLOCATE( lprpt )
    RETURN
2000 FORMAT( //10x, '==== 材 料 信 息 ====' )
2200 FORMAT( 12I5 )
2400 FORMAT( //2x, '材料类型序号', I5 / 2x, '单元类型序号', I5                    &
            / 2x, '材料作用因子', I5 )
2600 FORMAT( 2x, '自由度序列表', 10I5 / 14x, 10I5 )
2800 FORMAT( //2x, '错误: 材料', I5, '应输入单元类型', I5, '-', I5,              &
             /2x, '                ', 5x, '实输入单元类型', I5, '-', I5 )
    END
```

```
      SUBROUTINE RdFrc
!.....
!     模块功能
!         读写载荷信息.
!.....
      USE CtrlData
      USE LoadData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION fwork( 40 )

      IF( nerrc .GT. 0 ) RETURN
      CALL ShowMessage( '读载荷信息...' )

!.....读节点载荷.
      knodf = 0
      kedgf = 0
      kplnf = 0
      nnodg = mnodg - 1
      nnodp = mnodp - 1
      DO iload = 1, nload
        IF( nprnc .NE. 0 )                                              &
        WRITE( 12, 2000 ) iload
        READ( 11, 2200 ) nnodf, nedgf, nplnf
        loads( 1, iload ) = nnodf
        loads( 2, iload ) = nedgf
        loads( 3, iload ) = nplnf
        IF( nnodf .GT. 0 ) THEN
          IF( nprnc .NE. 0 ) WRITE( 12, 2400 )
          DO inodf = 1, nnodf
            knodf = knodf + 1
            READ( 11, 2600, ERR = 200 ) ipoin, itype,                   &
                  ( fwork( idofn ), idofn = 1, mdofn )
            IF( nprnc .NE. 0 )                                          &
            WRITE( 12, 2800 ) ipoin, itype, ( fwork( idofn ),           &
                                  idofn = 1, mdofn )
            IF( ipoin .LE. 0 .OR. ipoin .GT. npoin ) THEN
              WRITE( 12, 3000 ) ipoin
              nerrc = 843676
              RETURN
            END IF
            lnodf( 1, knodf ) = ipoin
            lnodf( 2, knodf ) = itype
            DO idofn = 1, mdofn
              vnodf( idofn, knodf ) = fwork( idofn )
            END DO
          END DO
        END IF
!.....读线分布载荷.
```

```
      IF( nedgf .GT. 0 ) THEN
        IF( nprnc .NE. 0 ) WRITE( 12, 3200 )
        DO iedgf = 1, nedgf
          ktype = 0
          kedgf = kedgf + 1
          WRITE( 12, 3400 ) iedgf
          DO inodg = 1, nnodg
            READ( 11, 2600, ERR = 400 ) ipoin, itype,                    &
                  ( fwork( idofn ), idofn = 1, mdofn )
            IF( nprnc .NE. 0 )                                             &
            WRITE( 12, 2800 ) ipoin, itype, ( fwork( idofn ),            &
                                idofn = 1, mdofn )
            IF( ipoin .GT. npoin ) THEN
              WRITE( 12, 3600 ) ipoin
              nerrc = 843676
              RETURN
            END IF
            IF( ktype .EQ. 0 ) THEN
              ktype = itype
            ELSE
              IF( itype .NE. 0 .AND. itype .NE. ktype ) THEN
                WRITE( 12, 3800 )
                nerrc = 54868
                RETURN
              END IF
            END IF
            ledgf( inodg, kedgf ) = ipoin
            DO idofn = 1, mdofn
              locat = ( inodg - 1 ) * mdofn + idofn
              vedgf( locat, kedgf ) = fwork( idofn )
            END DO
          END DO
          ledgf( mnodg, kedgf ) = ktype
        END DO
      END IF
!.....读面分布载荷.
      IF( nplnf .GT. 0 ) THEN
        IF( nprnc .NE. 0 ) WRITE( 12, 4000 )
        DO iplnf = 1, nplnf
          ktype = 0
          kplnf = kplnf + 1
          WRITE( 12, 3400 ) iplnf
          DO inodp = 1, nnodp
            READ( 11, 2600, ERR = 400 ) ipoin, itype,                    &
                  ( fwork( idofn ), idofn = 1, mdofn )
            IF( nprnc .NE. 0 )                                             &
            WRITE( 12, 2800 ) ipoin, itype, ( fwork( idofn ),            &
                                idofn = 1, mdofn )
```

```
                IF( ipoin .GT. npoin ) THEN
                  WRITE( 12, 4200 ) ipoin
                  nerrc = 843676
                  RETURN
                END IF
                IF( ktype .EQ. 0 ) THEN
                  ktype = itype
                ELSE
                  IF( itype .NE. 0 .AND. itype .NE. ktype ) THEN
                    WRITE( 12, 4400 )
                    nerrc = 54868
                    RETURN
                  END IF
                END IF
                lplnf( inodp, kplnf ) = ipoin
                DO idofn = 1, mdofn
                  locat = ( inodp - 1 ) * mdofn + idofn
                  vplnf( locat, kplnf ) = fwork( idofn )
                END DO
              END DO
              lplnf( mnodp, kplnf ) = ktype
            END DO
          END IF
        END DO
        RETURN
200     WRITE( 12, 4600 )
        nerrc = 1013
        RETURN
400     WRITE( 12, 4800 )
        nerrc = 1014
        RETURN
600     WRITE( 12, 5000 )
        nerrc = 1015
        RETURN
2000    FORMAT( //20x, ' 载荷工况 ', I5 )
2200    FORMAT( 3I5 )
2400    FORMAT( //10x, ' ==== 节 点 载 荷 ====  '//                    &
                    2x, ' 节点 类型          节 点 载 荷 值' )
2600    FORMAT( 2I5, 4F15.5 / ( 10X, 4F15.5 ) )
2800    FORMAT(   2x, 2I5, 3x, 4F15.5 / ( 15X, 4F15.5 ) )
3000    FORMAT( //2x, '致命错误: 节点力作用在无效的节点上 ', I5 )
3200    FORMAT( //10x, ' ==== 线 分 布 载 荷 ===='/                    &
                    2x, ' 节点 类型          分 布 载 荷 值' )
3400    FORMAT(   /20x, ' 序号', I5 )
3600    FORMAT( //2x, '致命错误: 线分布力作用在无效的节点上 ', I5 )
3800    FORMAT( //2x, '致命错误: 线分布力不可重复定义动载类型' )
4000    FORMAT( //10x, ' ==== 面 分 布 载 荷 ===='/                    &
```

```
                    2x,' 节点 类型          分 布 载 荷 值')
4200    FORMAT( //2x,'致命错误: 面分布力作用在无效的节点上 ', I5 )
4400    FORMAT( //2x,'致命错误: 面分布力不可重复定义动载类型' )
4600    FORMAT( // 2x,'错误: 读节点载荷出错' )
4800    FORMAT( // 2x,'错误: 读线分布载荷出错' )
5000    FORMAT( // 2x,'错误: 读面分布载荷出错' )
        END

        SUBROUTINE RDCnt
!.....
!       模块功能
!           读入集中质量, 集中刚度, 集中阻尼.
!.....
        USE CtrlData
        USE CentData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )

        IF( nerrc .GT. 0 ) RETURN
        CALL ShowMessage(  '读质点信息...' )

!.....读入集中质量.
        IF( nmssp .GT. 0 ) THEN
          IF( nprnc .NE. 0 ) WRITE( 12, 2000 )
          DO imssp = 1, nmssp
            READ(   11, 2200 ) lmssp( 1, imssp ), lmssp( 2, imssp ),          &
                               vmssp(      imssp )
            IF( nprnc .NE. 0 )                                                  &
            WRITE( 12, 2400 ) lmssp( 1, imssp ), lmssp( 2, imssp ),           &
                              vmssp(      imssp )
            IF( lmssp( 1, imssp ) .LE.       0 .OR.                           &
                lmssp( 1, imssp ) .GT. npoin .OR.                            &
                lmssp( 2, imssp ) .LE.       0 .OR.                           &
                lmssp( 2, imssp ) .GT. mdofn ) THEN
              nerrc = 2134
              RETURN
            END IF
          END DO
        END IF
!.....读入集中刚度.
        IF( nstfp .GT. 0 ) THEN
          IF( nprnc .NE. 0 ) WRITE( 12, 2600 )
          DO istfp = 1, nstfp
            READ(   11, 2200 ) lstfp( 1, istfp ), lstfp( 2, istfp ),          &
                               vstfp(      istfp )
            IF( nprnc .NE. 0 )                                                  &
            WRITE( 12, 2400 ) lstfp( 1, istfp ), lstfp( 2, istfp ),           &
                              vstfp(      istfp )
```

```
            IF( lstfp( 1, istfp ) .LE.        0 .OR.                    &
                lstfp( 1, istfp ) .GT. npoin .OR.                       &
                lstfp( 2, istfp ) .LE.        0 .OR.                     &
                lstfp( 2, istfp ) .GT. mdofn ) THEN
              nerrc = 2134
              RETURN
            END IF
          END DO
        END IF
!.....读入集中阻尼.
        IF( ndmpp .GT. 0 ) THEN
          IF( nprnc .NE. 0 ) WRITE( 12, 2800 )
          DO idmpp = 1, ndmpp
            READ(   11, 2200 ) ldmpp( 1, idmpp ), ldmpp( 2, idmpp ),    &
                               vdmpp(       idmpp )
            IF( nprnc .NE. 0 )                                            &
            WRITE( 12, 2400 ) ldmpp( 1, idmpp ), ldmpp( 2, idmpp ),     &
                               vdmpp(       idmpp )
            IF( ldmpp( 1, idmpp ) .LE.        0 .OR.                      &
                ldmpp( 1, idmpp ) .GT. npoin .OR.                         &
                ldmpp( 2, idmpp ) .LE.        0 .OR.                       &
                ldmpp( 2, idmpp ) .GT. mdofn ) THEN
              nerrc = 2134
              RETURN
            END IF
          END DO
        END IF
2000    FORMAT(/10x, '= = = = = 集 中 质 量 = = = = =' //                 &
                10x, ' 节点   自由度          集中质量' )
2200    FORMAT( 2I5, F15.5 )
2400    FORMAT( 10x, I5, 3x, I5, E15.5 )
2600    FORMAT(/10x, '= = = = = 集 中 刚 度 = = = = =' //                 &
                10x, ' 节点   自由度          集中刚度' )
2800    FORMAT(/10x, '= = = = = 集 中 阻 尼 = = = = =' //                 &
                10x, ' 节点   自由度          集中阻尼' )
        RETURN
        END

        SUBROUTINE RDSInt
        USE CtrlData
        USE MeshData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )

        IF( nerrc .GT. 0 ) RETURN
        CALL ShowMessage( '读斜约束信息...' )

        IF( nslnt .LE. 0 ) RETURN
```

```
      IF( nprnc .NE. 0 ) WRITE( 12, 1800 )
      DO islnt = 1, nslnt
        READ( 11, 2000 ) ipoin, idofn, ( vslnt( jdofn, islnt ),          &
                         jdofn = 1, mdofn )
        lslnt( 1, islnt ) = ipoin
        lslnt( 2, islnt ) = idofn
        IF( ipoin .LT.-ndime .OR. ipoin .EQ. 0 .OR.                      &
            ipoin .GT. npoin ) THEN
          WRITE( 12, 2200 ) ipoin
          nerrc = 10213
          RETURN
        END IF
        IF( idofn .LE. 0 .OR. idofn .GT. mdofn ) THEN
          WRITE( 12, 2400 ) idofn
          nerrc = 12543
          RETURN
        END IF
        IF( nprnc .NE. 0 )                                               &
        WRITE( 12, 2600 ) ipoin, idofn, ( vslnt( jdofn, islnt ),         &
                                  jdofn = 1, mdofn )
      END DO
1800  FORMAT( // 10X, '= = = = = 广 义 自 由 度 = = = = = ' //           &
                  1X, '节点   自由度          自由度转化系数')
2000  FORMAT( 2I5, 4F15.5 / ( 10X, 4F15.5 ) )
2200  FORMAT( 1X, '错误: 广义自由度对应无效的节点', I5 )
2400  FORMAT( 1X, '错误: 广义自由度对应无效的主自由度', I5 )
2600  FORMAT( I5, 3X, I4, 3X, 3F15.5 / ( 15X, 3F15.5 ) )
      RETURN
      END

      SUBROUTINE RdOptAxis
      USE CtrlData
      USE MeshData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION axist( 3 )
      const = 0.0D0
      IF( nerrc .GT. 0 ) RETURN
      READ( 11, '( 3F15.5 )' ) basew
      DO idime = 1, ndime
        READ( 11, '( 3F15.5 )' ) axist
        DO jdime = 1, ndime
      axisw( idime, jdime ) = axist( jdime )
          const = const + DABS( axist( jdime ) )
        END DO
      END DO
      IF( const .LT. 1.0D-8 ) THEN
        DO idime = 1, ndime
          axisw( idime, idime ) = 1.0D0
```

```
      END DO
    END IF
    axist( 1 ) = 0.0D0
    axist( 2 ) = 0.0D0
    axist( 3 ) = 0.0D0
    IF( nprnc .NE. 0 ) THEN
      DO idime = 1, ndime
        axist( idime ) = basew( idime )
      END DO
      WRITE( 12, 2000 ) axist
      DO idime = 1, ndime
        DO jdime = 1, ndime
          axist( jdime ) = axisw( idime, jdime )
        END DO
        WRITE( 12, 2200 ) idime, axist
      END DO
    END IF
2000 FORMAT( // 15X, '优化中心和优化轴线信息',                                    &
             //   2X, '优化中心:', 3F15.5 )
2200 FORMAT(        2X, '第', I2, '轴线:', 3F15.5 )
    END

    SUBROUTINE AutoLink( strng )
!.....
!   模块功能:
!       坐标相同的节点自动连接.
!.....
    USE CtrlData
    USE MeshData
    CHARACTER * 80 strng
    READ( strng( 11:20 ), 2000, ERR = 200 ) vskip
    IF( vskip .LT. 1.0D-5 ) vskip = 1.0D-5

    DO ipoin = 1, npoin
      IF( ipoin .EQ. llnks( ipoin ) ) THEN
        DO jpoin = ipoin + 1, npoin
          IF( jpoin .EQ. llnks( jpoin ) ) THEN
            displ = DABS( coord(1, ipoin) - coord(1, jpoin) )
            IF( displ .LT. vskip ) THEN
              nflag = 0
              DO idime = 2, ndime
                displ = DABS( coord( idime, ipoin ) -                          &
                              coord( idime, jpoin ) )
                IF( displ .GT. vskip ) nflag = 1
              END DO
              IF( nflag .EQ. 0 ) llnks( jpoin ) = ipoin
            END IF
          END IF
```

```
            END DO
          END IF
        END DO
        RETURN
200     nerrc = 32435
        WRITE( 12, 2200 )
        RETURN
2000    FORMAT( F10.5 )
2200    FORMAT( 2x, // ' 错误的自动连接宏命令!' // )
        END
```

附 3 Execute.f90 文本

```
      SUBROUTINE Execute
!.....
!     模块功能
!       主子模块,负责带宽优化,设定波阵信息,读入宏命令,并执行各条
!     宏命令.
!.....
      USE CtrlData
      USE MeshData
      USE LoadData
      USE CentData
      USE GlobData
      USE ElmtData
      USE SolvData
      USE ExtraMesh
      USE FrontData
      USE MacroData
      USE FactorData
      USE EarthQuake
      USE ReSolveData
      IF( nerrc .GT. 0 ) RETURN
      CALL ShowMessage( '开设动态数组' )
!.....设定参数.
      CALL InitGlob
      CALL InitElmt
      CALL InitSolv
      CALL InitMacro
      CALL InitFactor
      CALL InitResolveData

      CALL NodDof
      CALL WavOpt
      CALL SlvOpt
      CALL InitWave
      CALL SetRec
      CALL ReadMacro
      CALL GetExtraMesh
      CALL ShowPhase( kload )
      CALL ShowGraph( 0.0D0, 0.0D0, 0 )

      IF( nincc .EQ. 0 ) nincc = 1
!.....依次执行各条宏命令.
      DO WHILE( nexec .EQ. 1 .AND. nerrc .EQ. 0 )
        CALL ShowMacro
```

```
        ncomm = lmacr( 1, kmacr )
        IF( ncomm .LT.    0 .OR.   ncomm .GT. 140 ) nerrc = 3585
        IF( ncomm .GE.    0 .AND. ncomm .LE.   20 ) CALL Exec020
        IF( ncomm .GT.   20 .AND. ncomm .LE.   40 ) CALL Exec040
        IF( ncomm .GT.   40 .AND. ncomm .LE.   60 ) CALL Exec060
        IF( ncomm .GT.   60 .AND. ncomm .LE.   80 ) CALL Exec080
        IF( ncomm .GT.   80 .AND. ncomm .LE. 120 ) CALL Exec120
        IF( ncomm .GT. 120 .AND. ncomm .LE. 140 ) CALL Exec140
      END DO

      CALL DelWave
      CALL DelGlob
      CALL DelElmt
      CALL DelSolv
      CALL DelMacro
      CALL DelExtraMesh
      CALL DelEarthQuake
      RETURN
      END

      SUBROUTINE Exec020
!.....
!     模块功能
!         本模块负责执行命令编号在 0-20 之间的各条宏命令.
!.....
      USE CtrlData
      USE GlobData
      USE MacroData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      CHARACTER * 60 strng

      ncomm = lmacr( 1, kmacr )
      para1 = pmacr( 1, kmacr )
      para2 = pmacr( 2, kmacr )
      para3 = pmacr( 3, kmacr )
      para4 = pmacr( 4, kmacr )
      para5 = pmacr( 5, kmacr )

      IF( ncomm .EQ. 0 ) THEN
!.....结束宏命令.
        nexec = 0
      ELSE IF( ncomm .EQ. 1 ) THEN
!.....宏命令 loop
        nendl                  = lmacr( 3, kmacr )
        pmacr( 1, nendl ) = pmacr( 1, kmacr )
        pmacr( 2, nendl ) = pmacr( 2, kmacr )
        pmacr( 3, nendl ) = pmacr( 3, kmacr )
        kmacr = lmacr( 2, kmacr )
```

```
            kaitk = 0
          ELSE IF( ncomm .EQ. 2 ) THEN
!.....宏命令 endloop
            dstep = pmacr( 3, kmacr )
            pmacr( 1, kmacr ) = pmacr( 1, kmacr ) + dstep
            dflag = pmacr( 1, kmacr ) - pmacr( 2, kmacr )
            dflag = dflag * dstep / DABS( dstep )
            IF( dflag .GT. dstep * 1.0D-5 ) THEN
              kmacr = kmacr + 1
            ELSE
              kmacr = lmacr( 2, kmacr )
            END IF
          ELSE IF( ncomm .EQ. 3 ) THEN
!.....宏命令 tolen
            IF( para1 .GE. 1.0D0 ) THEN
              para1 =-para1
              para1 = 10.0D0 ** para1
            END IF
            IF( para1 .LT. 1.0D-12 ) para1 = 1.0D-5
            IF( para2 .LT. 1.0D-12 ) para2 = para1
!.....以下为新版误差控制
            wtolc = GetError()
            CALL ShowError( para1, wtolc )
            IF( necho .EQ. 1 ) THEN
              WRITE( strng, 2000 ) dtolc
              CALL ShowMessage( strng )
            END IF
            IF( wtolc .LT. para1 ) THEN
!.....达到足够的精度.
              kmacr = lmacr( 2, kmacr )
            ELSE
!.....未达到足够的精度.
              kmacr = kmacr + 1
            END IF
            RETURN
!.....以下为旧版误差控制
            dtolm = 1.0D0 / 10.D0 ** npar1
            ftolm = 1.0D0 / 10.D0 ** npar2
            IF( nincc .LE. 1 .OR. nincc .EQ. 3 ) THEN
              IF( necho .EQ. 1 ) THEN
                WRITE( strng, 2000 ) dtolc
                CALL ShowMessage( strng )
              END IF
              IF( dtolc .LT. para1 ) THEN
!.....达到足够的精度.
                kmacr = lmacr( 2, kmacr )
              ELSE
```

```
!.....未达到足够的精度.
                kmacr = kmacr + 1
              END IF
            ELSE
              IF( necho .EQ. 1 ) THEN
                WRITE( strng, 2200 ) dtolc, ftolc
                CALL ShowMessage( strng )
              END IF
              IF( dtolc .LT. para1 .AND. ftolc .LT. para2 ) THEN
!.....达到足够的精度.
                kmacr = lmacr( 2, kmacr )
              ELSE
!.....未达到足够的精度.
                kmacr = kmacr + 1
              END IF
            END IF
          ELSE IF( ncomm .EQ. 4 ) THEN
!.....宏命令 nottolen
            IF( para1 .GE. 1.0D0 ) THEN
              para1 =-para1
              para1 = 10.0D0 ** para1
            END IF
            IF( para1 .LT. 1.0D-12 ) para1 = 1.0D-5
            IF( para2 .LT. 1.0D-12 ) para2 = para1
!.....以下为新版误差控制
            wtolc = GetError()
            CALL ShowError( para1, wtolc )
            IF( necho .EQ. 1 ) THEN
              WRITE( strng, 2000 ) dtolc
              CALL ShowMessage( strng )
            END IF
            IF( wtolc .LT. para1 ) THEN
!.....达到足够的精度.
              kmacr = kmacr + 1
            ELSE
!.....未达到足够的精度.
              kmacr = lmacr( 2, kmacr )
            END IF
            RETURN
!.....以下为旧版误差控制
            IF( nincc .LE. 1 .OR. nincc .EQ. 3 ) THEN
              IF( necho .EQ. 1 ) THEN
                WRITE( strng, 2000 ) dtolc
                CALL ShowMessage( strng )
              END IF
              IF( dtolc .LT. para1 ) THEN
!.....达到足够的精度.
```

```
                kmacr = kmacr + 1
              ELSE
!.....未达到足够的精度.
                kmacr = lmacr( 2, kmacr )
              END IF
            ELSE
              IF( necho .EQ. 1 ) THEN
                WRITE( strng, 2200 ) dtolc, ftolc
                CALL ShowMessage( strng )
              END IF
              IF( dtolc .LT. para1 .AND. ftolc .LT. para2 ) THEN
!.....达到足够的精度.
                kmacr = kmacr + 1
              ELSE
!.....未达到足够的精度.
                kmacr = lmacr( 2, kmacr )
              END IF
            END IF
          ELSE IF( ncomm .EQ. 5 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL ShowMessage( '运行暂停，敲回车键继续...' )
            READ( *, * )
          ELSE
            WRITE( 12, 8000 ) ncomm
            nerrc = 45496
          END IF
          RETURN
2000      FORMAT( '位移增量二模', E15.6 )
2200      FORMAT( '位移增量二模', E15.6, '，残差二模', E15.6 )
8000      FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
          END

          SUBROUTINE Exec040
!.....
!         模块功能
!             本模块负责执行命令编号在 21-40 之间的各条宏命令.
!.....
          USE CtrlData
          USE MeshData
          USE CentData
          USE LoadData
          USE GlobData
          USE ElmtData
          USE SolvData
          USE FrontData
          USE MacroData
          USE ExtraMesh
```



数据加载失败，请稍后重试！

```
            nshfc = 0
            kmacr = lmacr( 2, kmacr )
            IF( nmssc .NE. 1 ) RETURN
            IF( nmssc .EQ. 1 ) ldecc( 2 ) = 1
            CALL ElmLib( diagm, fwork, iswth )
          ELSE IF( ncomm .EQ. 25 ) THEN
!.....宏命令 force
            ngaus = 3
            nfrcc = 1
            iswth = 6
            kmacr = lmacr( 2, kmacr )
            IF( npar1 .GT. 0 ) kload = npar1
            IF( kload .GT. nload ) THEN
              nerrc = 1582
              RETURN
            END IF
            CALL ShowPhase( kload )
            CALL ElmLib( fwork, forcs, iswth )
            CALL Load( forcs )
            IF( nearc .EQ. 1 ) THEN
              CALL QuakeForce
              DO idofs = 1, ndofs
                forcs( idofs ) = forcs( idofs ) + frceq( idofs )
              END DO
            END IF
            CALL DynFrc
          ELSE IF( ncomm .EQ. 26 ) THEN
!.....宏命令 react
            iswth = 10
            IF( nreac .EQ. 0 ) CALL ElmLib( fwork, react, iswth )
            CALL PrntDsp( npar1, npar2, npar3, 3 )
            kmacr = lmacr( 2, kmacr )
            nreac = 1
          ELSE IF( ncomm .EQ. 30 .OR. ncomm .EQ. 31 ) THEN
!.....宏命令 stress,nodestrs
            kmacr = lmacr( 2, kmacr )
            IF( npar2 .EQ. 0 ) npar2 = npar1
            IF( npar1 .EQ. 0 ) npar1 = 1
            IF( npar3 .EQ. 0 ) npar3 = 1
            IF( ncomm .EQ. 30 ) THEN
              iswth = 7
              WRITE( 12, 2000 )
              IF( npar2 .EQ. 0 ) npar2 = nelem
              ntmp1 = npar1
              ntmp2 = npar2
              ntmp3 = npar3
            ELSE
              iswth = mswth + 7
```

```
            ntmp1 = 1
            ntmp3 = 1
            ntmp2 = nelem
            WRITE( 12, 2200 )
            IF( npar2 .EQ. 0 ) npar2 = npoin
            CALL InitFloat( strss, 7 * npoin, 0.0D0 )
            CALL InitFloat( strns, 7 * npoin, 0.0D0 )
            IF( nnodx .NE. 0 ) THEN
              CALL InitFloat( strsx, 7 * nnodx, 0.0D0 )
              CALL InitFloat( strnx, 7 * nnodx, 0.0D0 )
            END IF
          END IF
          DO ielem = ntmp1, ntmp2, ntmp3
            CALL ElmOpt( ielem, iswth )
            IF( nerrc .GT. 0 ) RETURN
          END DO
          IF( ncomm .EQ. 31 ) CALL PrntStrs( npar1, npar2, npar3 )
        ELSE IF( ncomm .EQ. 39 ) THEN
!.....宏命令 liquid
          iswth = 12
          CALL ElmLib( fwork, react, iswth )
          kmacr = lmacr( 2, kmacr )
        ELSE
          WRITE( 12, 8000 ) ncomm
          nerrc = 45496
        END IF
        RETURN
2000    FORMAT( // 2x, 60( '=' ) // 12x, '高　斯　点　应　力',                &
                // 2x, 60( '=' ) // )
2200    FORMAT( // 2x, 60( '=' ) // 12x, '节　点　应　力',                    &
                // 2x, 60( '=' ) // )
8000    FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
        END

        SUBROUTINE Exec060
!.....
!       模块功能
!           本模块负责执行命令编号在 81-100 之间的各条宏命令.
!.....
        USE CtrlData
        USE MeshData
        USE CentData
        USE LoadData
        USE GlobData
        USE SolvData
        USE ExtraMesh
        USE FrontData
```

```
      USE MacroData
      USE ReSolveData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )

      ncomm = lmacr( 1, kmacr )
      para1 = pmacr( 1, kmacr )
      para2 = pmacr( 2, kmacr )
      para3 = pmacr( 3, kmacr )
      para4 = pmacr( 4, kmacr )
      para5 = pmacr( 5, kmacr )
      npar1 = pmacr( 1, kmacr ) + 0.5D0
      npar2 = pmacr( 2, kmacr ) + 0.5D0
      npar3 = pmacr( 3, kmacr ) + 0.5D0
      npar4 = pmacr( 4, kmacr ) + 0.5D0
      npar5 = pmacr( 5, kmacr ) + 0.5D0
      IF( ncomm .EQ. 41 ) THEN
!.....宏命令 solve
        kmacr = lmacr( 2, kmacr )
        IF( nfrcc .EQ. 0 ) nerrc = 1455
        IF( ldecc( 1 ) .EQ. 0 ) nerrc = 5001
        IF( ntypc .EQ. 2 .AND. nshfc .EQ. 0 ) nerrc = 5001
        IF( nerrc .GT. 0 ) RETURN
!.....更新迭代步
        DO idofs = 1, ndofs
          dspis( idofs, kload ) = disps( idofs, kload )
        END DO
!.....参数设定
        nfrcc = 0
        nupdc = 0
        nrsdc = 0
        nreac = 0
        iswth = 13
        IF( ldecc( 1 ) .EQ. 1 ) THEN
          ldecc( 1 ) = 2
          CALL DealFix( 1 )
          CALL DeComp( 1, npar1 )
        END IF
        CALL BckSub( 1 )
        CALL LinkNode
        IF( npar2 .EQ. 1 ) CALL Aitken( vslvs )
!.....自由度光滑处理和交换
        IF( nsmts .GT. 0 ) CALL Smooth
        CALL Exchang( 1 )
        CALL GetExtraDisp
        CALL BackSolveResult( 1 )
      END IF
      IF( ncomm .EQ. 42 ) THEN
!.....宏命令 shift
```

```
            kflag =-1
            IF( para1 .LT. 0 ) kflag = 1
            IF( npar2 .EQ. 0 ) THEN
              para1 = para1 * 8.0D0 * DATAN( 1.0D0 )
              para1 = para1 * para1
            END IF
            shift( 1 ) = 1.0D0
            shift( 2 ) = para1 * kflag
            DO ishft = 3, 10
              shift( ishft ) = 0.0D0
            END DO
            DO imatx = 1, 2
              IF( lsavc( imatx ) .NE. 0 .AND.                                    &
                  ldecc( imatx ) .NE. 1 ) THEN
                nerrc = 2334
                RETURN
              END IF
            END DO
            CALL ShftMat
            kmacr = lmacr( 2, kmacr )
          END IF
          IF( ncomm .EQ. 43 ) THEN
!.....宏命令 Wilson
            ndync = 1
            CALL Wilson
            kmacr = lmacr( 2, kmacr )
          END IF
          IF( ncomm .EQ. 44 ) THEN
!.....宏命令 Newmark
            ndync = 2
            CALL Newmark
            kmacr = lmacr( 2, kmacr )
          END IF
          IF( ncomm .EQ. 45 ) THEN
!.....宏命令 eigenpair
            iswth = 1
            kmacr = lmacr( 2, kmacr )
            IF( para1 .LE. 1.0D-12 ) para1 = 1.0D-5
            IF( ldecc( 1 ) .EQ. 0 ) THEN
              WRITE( 12, 2000 )
              nerrc = 1324
              RETURN
            END IF
            IF( ldecc( 2 ) .NE. 1 ) THEN
              WRITE( 12, 2200 )
              nerrc = 1325
              RETURN
            END IF
```

```
            IF( ntypc .GT. 3 .AND. ldecc( 3 ) .NE. 1 ) THEN
              WRITE( 12, 2400 )
              nerrc = 1326
              RETURN
            END IF
            IF( ldecc( 1 ) .EQ. 1 ) THEN
              index = 0
              ldecc( 1 ) = 2
              CALL DealFix( 1 )
              CALL DeComp( 1, index )
            END IF
            CALL InitFloat( fstrs, ndofs, 0.0D0 )
            CALL EigenP( para1, npar2 )
            CALL GetExtraDisp
          END IF
          IF( ncomm .EQ. 46 ) THEN
!.....宏命令 accel
            IF( nfrcc .EQ. 0 ) nerrc = 36509
            IF( nerrc .GT. 0 ) RETURN
            nfrcc = 0
            nupdc = 0
            kmacr = lmacr( 2, kmacr )
            IF( lsavc( 3 ) .NE. 0 ) THEN
              iswth = 3
              IF( ldecc( 3 ) .NE. 1 ) nerrc = 213256
              IF( nerrc .GT. 0 ) RETURN
              DO idofs = 1, ndofs
                vslvs( idofs ) = -vslvs( idofs )
              END DO
              CALL DotMat( veloc, vslvs, iswth, kload, nload )
              IF( nerrc .GT. 0 ) RETURN
              DO idofs = 1, ndofs
                vslvs( idofs ) = -vslvs( idofs )
              END DO
            END IF
            DO idofs = 1, ndofs
              accel( idofs, kload ) = 0.0D0
            END DO
            iswth = 2
            IF( ldecc( 2 ) .EQ. 0 ) THEN
              nerrc = 435467
              RETURN
            END IF
            IF( ldecc( 2 ) .EQ. 1 ) THEN
              index = 0
              ldecc( 2 ) = 2
              CALL DealFix( 2 )
              CALL DeComp( 2, index )
```

```
            END IF
            CALL BckSub( 2 )
            CALL Exchang( 2 )
            IF( npar1 .NE. 0 ) CALL PrntDsp( 1, npoin, 1, 2 )
          END IF
          IF( ncomm .EQ. 47 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL OptStep( npar1, npar2 )
          END IF
          IF( ncomm .EQ. 48 ) THEN
!.....宏命令 initial
            kmacr = lmacr( 2, kmacr )
            CALL Initial
          END IF
          IF( ncomm .EQ. 49 ) THEN
!.....宏命令 update
            kmacr = lmacr( 2, kmacr )

            IF( nhstr .GT. 0 ) CALL ElmLib( fwork, react, 11 )
            CALL Update
            CALL BackSolveResult( 2 )
            CALL ShowTime( timec )
          END IF
          IF( ncomm .EQ. 50 ) THEN
!.....宏命令 Resolve
            kmacr = lmacr( 2, kmacr )
            CALL ReSolveStif
            CALL ReDecomp
            CALL ReBakSub
            CALL ReExchang
          END IF
          IF( ncomm .EQ. 51 ) THEN
!.....宏命令 crank
            ndync = 3
            CALL Crank
            kmacr = lmacr( 2, kmacr )
          END IF
          IF( ncomm .GT. 51 ) THEN
            WRITE( 12, 8000 ) ncomm
            nerrc = 45496
          END IF
          RETURN
2000      FORMAT( //2x, '错误: 特征对分析前先计算刚度阵！' )
2200      FORMAT( //2x, '错误: 特征对分析前先计算质量阵！' )
2400      FORMAT( //2x, '错误: 特征对分析前先计算阻尼阵！' )
8000      FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
          END
```

```
      SUBROUTINE Exec080
!.....
!     模块功能
!         本模块负责执行命令编号在 61-80 之间的各条宏命令.
!.....
      USE CtrlData
      USE MeshData
      USE CentData
      USE LoadData
      USE GlobData
      USE MacroData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      CHARACTER * 60 strng

      ncomm = lmacr( 1, kmacr )
      para1 = pmacr( 1, kmacr )
      para2 = pmacr( 2, kmacr )
      para3 = pmacr( 3, kmacr )
      para4 = pmacr( 4, kmacr )
      para5 = pmacr( 5, kmacr )
      npar1 = pmacr( 1, kmacr ) + 0.5D0
      npar2 = pmacr( 2, kmacr ) + 0.5D0
      npar3 = pmacr( 3, kmacr ) + 0.5D0
      npar4 = pmacr( 4, kmacr ) + 0.5D0
      npar5 = pmacr( 5, kmacr ) + 0.5D0
      IF( ncomm .EQ. 61 ) THEN
!.....宏命令 printdisp
        kmacr = lmacr( 2, kmacr )
        CALL PrntDsp( npar1, npar2, npar3, 0 )
      ELSE IF( ncomm .EQ. 62 ) THEN
!.....宏命令 printvel
        kmacr = lmacr( 2, kmacr )
        IF( ntypc .EQ. 2 )                                          &
        CALL PrntDsp( npar1, npar2, npar3, 1 )
      ELSE IF( ncomm .EQ. 63 ) THEN
!.....宏命令 printacc
        kmacr = lmacr( 2, kmacr )
        IF( ntypc .EQ. 2 )                                          &
        CALL PrntDsp( npar1, npar2, npar3, 2 )
      ELSE IF( ncomm .EQ. 64 ) THEN
!.....宏命令 plotmesh
        kmacr = lmacr( 2, kmacr )
        CALL PrntPlt( 0, npar1 )
      ELSE IF( ncomm .EQ. 65 ) THEN
!.....宏命令 trace
        kmacr = lmacr( 2, kmacr )
```

```
            IF( ntypc .EQ. 0 ) THEN
              IF( npar1 .GT. 1 ) nerrc = 3294
              IF( nerrc .GT. 0 ) RETURN
              CALL Trace( npar1, npar2, npar3 )
            ELSE IF( ntypc .EQ. 2 ) THEN
              CALL Trace( npar1, npar2, npar3 )
            END IF
          ELSE IF( ncomm .EQ. 66 ) THEN
!.....宏命令 echoon
            kmacr = lmacr( 2, kmacr )
            necho = 1
          ELSE IF( ncomm .EQ. 67 ) THEN
!.....宏命令 echooff
            kmacr = lmacr( 2, kmacr )
            necho = 0
          ELSE IF( ncomm .EQ. 68 ) THEN
!.....宏命令 echodisp
            kmacr = lmacr( 2, kmacr )
            IF( npar1 .GT. 0 .AND. npar1 .LE. npoin ) THEN
              IF( npar2 .GT. 0 .AND. npar2 .LE. mdofn ) THEN
                idofs = ( npar1 - 1 ) * mdofn + npar2
                WRITE( strng, 2000 ) npar1, npar2, disps( idofs, kload )
              END IF
            END IF
          ELSE IF( ncomm .EQ. 69 ) THEN
!.....宏命令 echovelo
            kmacr = lmacr( 2, kmacr )
            IF( npar1 .GT. 0 .AND. npar1 .LE. npoin ) THEN
              IF( npar2 .GT. 0 .AND. npar2 .LE. mdofn ) THEN
                idofs = ( npar1 - 1 ) * mdofn + npar2
                WRITE( strng, 2200 ) npar1, npar2, veloc( idofs, kload )
                CALL ShowMessage( strng )
              END IF
            END IF
          ELSE IF( ncomm .EQ. 70 ) THEN
!.....宏命令 echoacc
            kmacr = lmacr( 2, kmacr )
            IF( npar1 .GT. 0 .AND. npar1 .LE. npoin ) THEN
              IF( npar2 .GT. 0 .AND. npar2 .LE. mdofn ) THEN
                idofs = ( npar1 - 1 ) * mdofn + npar2
                WRITE( strng, 2400 ) npar1, npar2, accel( idofs, kload )
                CALL ShowMessage( strng )
              END IF
            END IF
          ELSE IF( ncomm .EQ. 71 ) THEN
!.....宏命令 echotime
            kmacr = lmacr( 2, kmacr )
```

```
            IF( DABS( para1 ) .LT. 1.0D-12 ) para1 = 1.0D0
            WRITE( strng, 2600 ) timec / para1
            CALL ShowMessage( strng )
          ELSE IF( ncomm .EQ. 72 ) THEN
!.....宏命令 echofctor
            kmacr = lmacr( 2, kmacr )
            WRITE( strng, 2800 ) fctor
            CALL ShowMessage( strng )
          ELSE IF( ncomm .EQ. 73 ) THEN
!.....宏命令 echotol
            kmacr = lmacr( 2, kmacr )
            WRITE( strng, 3000 ) dtolc, ftolc
            CALL ShowMessage( strng )
          ELSE IF( ncomm .EQ. 75 ) THEN
!.....宏命令 plotdisp
            kmacr = lmacr( 2, kmacr )
            CALL PrntPlt( 1, npar1 )
          ELSE IF( ncomm .EQ. 76 ) THEN
!.....宏命令 plotstrs
            kmacr = lmacr( 2, kmacr )
            CALL PrntPlt( 2, npar1 )
          ELSE IF( ncomm .EQ. 77 ) THEN
!.....宏命令 printstat
            kmacr = lmacr( 2, kmacr )
            CALL PrntStat
          ELSE IF( ncomm .EQ. 78 ) THEN
!.....宏命令 plotstate
            kmacr = lmacr( 2, kmacr )
            CALL PlotStat
          ELSE IF( ncomm .EQ. 79 ) THEN
            idofs = ( npar2 - 1 ) * mdofn + npar3
            IF( npar2 .LE. 0 .OR. npar2 .GT. npoin ) THEN
              nerrc = 469578
              RETURN
            END IF
            IF( npar3 .LE. 0 .OR. npar3 .GT. mdofn ) THEN
              nerrc = 469578
              RETURN
            END IF
            IF( npar1 .EQ. 0 .OR. npar1 .EQ. 1 ) THEN
              value = disps( idofs, kload )
            ELSE IF( npar2 .EQ. 2 ) THEN
              value = veloc( idofs, kload )
            ELSE
              value = accel( idofs, kload )
            END IF
            CALL ShowShown( npar1, npar2, npar3 )
```

```
            CALL ShowGraph( timec, value, 1 )
            kmacr = lmacr( 2, kmacr )
          ELSE IF( ncomm .EQ. 80 ) THEN
            idofs = ( npar2 - 1 ) * mdofn + npar3
            IF( npar1 .EQ. 0 .OR. npar1 .EQ. 1 ) THEN
              value = disps( idofs, kload )
            ELSE IF( npar2 .EQ. 2 ) THEN
              value = veloc( idofs, kload )
            ELSE
              value = accel( idofs, kload )
            END IF
            CALL ShowShown( npar1, npar2, npar3 )
            CALL ShowGraph( kload, value, 1 )
            kmacr = lmacr( 2, kmacr )
          ELSE
            WRITE( 12, 8000 ) ncomm
            nerrc = 45496
          END IF
          RETURN
2000      FORMAT( '第', I5, '节点第', I2, '自由度位移为', G15.6 )
2200      FORMAT( '第', I5, '节点第', I2, '自由度速度为', G15.6 )
2400      FORMAT( '第', I5, '节点第', I2, '自由度加速度为', G15.6 )
2600      FORMAT( '当步时间为：', G15.6 )
2800      FORMAT( '载荷因子为：', G15.6 )
3000      FORMAT( '位移增量二模', E15.6, '，内力残差二模', E15.6 )
8000      FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
          END

          SUBROUTINE Exec120
!.....
!         模块功能
!             本模块负责执行命令编号在 81-120 之间的各条宏命令.
!.....
          USE CtrlData
          USE MeshData
          USE CentData
          USE LoadData
          USE GlobData
          USE MacroData
          USE FactorData
          USE EarthQuake
          IMPLICIT DOUBLE PRECISION( a-h, o-z )

          ncomm = lmacr( 1, kmacr )
          para1 = pmacr( 1, kmacr )
          para2 = pmacr( 2, kmacr )
          para3 = pmacr( 3, kmacr )
```

```
            para4 = pmacr( 4, kmacr )
            para5 = pmacr( 5, kmacr )
            npar1 = pmacr( 1, kmacr ) + 0.5D0
            npar2 = pmacr( 2, kmacr ) + 0.5D0
            npar3 = pmacr( 3, kmacr ) + 0.5D0
            npar4 = pmacr( 4, kmacr ) + 0.5D0
            npar5 = pmacr( 5, kmacr ) + 0.5D0
            SELECT CASE( ncomm )
              CASE( 81 )
!.....宏命令 timestep
                kmacr = lmacr( 2, kmacr )
                tstep = para1
              CASE( 82 )
!.....宏命令 loadcase
                kload = npar1
                CALL ShowPhase( kload )
                kmacr = lmacr( 2, kmacr )
              CASE( 83 )
!.....宏命令 nextload
                kload = kload + 1
                CALL ShowPhase( kload )
                kmacr = lmacr( 2, kmacr )
              CASE( 84 )
!.....宏命令 alpha
                kmacr = lmacr( 2, kmacr )
                alpha = para1
              CASE( 85 )
!.....宏命令 belta
                kmacr = lmacr( 2, kmacr )
                belta = para1
              CASE( 86 )
!.....宏命令 theta
                kmacr = lmacr( 2, kmacr )
                theta = para1
              CASE( 87 )
!.....宏命令 dispstep
                kmacr = lmacr( 2, kmacr )
                dstep = para1
              CASE( 88 )
!.....宏命令 lockdof
                kmacr = lmacr( 2, kmacr )
!               CALL LckDOF( npar2, kdofc, npar1 )
              CASE( 89 )
!.....宏命令 lockpromp
                kmacr = lmacr( 2, kmacr )
                CALL LockPromp( npar1 )
              CASE( 90 )
```

```
!.....宏命令 prompt
                kmacr = lmacr( 2, kmacr )
                CALL SetPromp( npar1, para2 )
              CASE( 91 )
!.....宏命令 quake
                CALL ReadQuake
                kmacr = lmacr( 2, kmacr )
              CASE( 92 )
!.....宏命令 normalize
                nuntc = 1
                kmacr = lmacr( 2, kmacr )
              CASE( 93 )
!.....宏命令 lowlimit
                IF( npar1 .GT. mdofn ) nerrc = 35345
                IF( npar1 .LT. 0 ) nerrc = 35345
                IF( nerrc .GT. 0 ) RETURN
                llmts( 1, npar1 ) = 1
                kmacr = lmacr( 2, kmacr )
                vlmts( 1, npar1 ) = para2
              CASE( 94 )
!.....宏命令 uplimit
                IF( npar1 .GT. mdofn ) nerrc = 35345
                IF( npar1 .LT. 0 ) nerrc = 35345
                IF( nerrc .GT. 0 ) RETURN
                llmts( 2, npar1 ) = 1
                kmacr = lmacr( 2, kmacr )
                vlmts( 2, npar1 ) = para2
              CASE( 95 )
!.....宏命令 smooth
                IF( npar1 .GT. mdofn ) nerrc = 35345
                IF( npar1 .LE. 0 ) nerrc = 35345
                IF( nerrc .GT. 0 ) RETURN
                lsmts( npar1 ) = 1
                nsmts = 1
                kmacr = lmacr( 2, kmacr )
              CASE( 96 )
!.....宏命令 construct
                kmacr = lmacr( 2, kmacr )
                CALL UpdateStructure
                IF( nerrc .NE. 0 ) RETURN
                IF( npar1 .EQ. 0 ) RETURN
                IF( npar3 .EQ. 0 ) npar3 = 1
                IF( npar2 .EQ. 0 ) npar2 = npar1
                DO ielem = npar1, npar2, npar3
                  luses( ielem ) = 1
                END DO
              CASE( 97 )
```

```
!.....宏命令 destruct
            kmacr = lmacr( 2, kmacr )
            CALL UpdateStructure
            IF( nerrc .NE. 0 ) RETURN
            IF( npar1 .EQ. 0 ) RETURN
            IF( npar3 .EQ. 0 ) npar3 = 1
            IF( npar2 .LE. 0 ) npar2 = npar1
            DO ielem = npar1, npar2, npar3
              luses( ielem ) = 0
              DO ipres = 1, nstrh
                strsh( ipres, ielem ) = 0.0D0
              END DO
            END DO
          CASE( 98 )
!.....宏命令 free
            kmacr = lmacr( 2, kmacr )
            CALL UpdateStructure
            IF( nerrc .NE. 0 ) RETURN
            IF( npar2 .EQ. 0 ) RETURN
            IF( npar1 .LE. 0 .OR. npar1 .GT. mdofn ) RETURN
            IF( npar3 .EQ. 0 ) npar3 = npar2
            IF( npar4 .EQ. 0 ) npar4 = 1
            DO ipoin = npar2, npar3, npar4
              lfixs( npar1, ipoin ) = 0
            END DO
          CASE( 99 )
!.....宏命令 fix
            kmacr = lmacr( 2, kmacr )
            CALL UpdateStructure
            IF( nerrc .NE. 0 ) RETURN
            IF( npar2 .EQ. 0 ) RETURN
            IF( npar1 .LE. 0 .OR. npar1 .GT. mdofn ) RETURN
            IF( npar3 .EQ. 0 ) npar3 = npar2
            IF( npar4 .EQ. 0 ) npar4 = 1
            DO ipoin = npar2, npar3, npar4
              lfixs( npar1, ipoin ) = 1
            END DO
          CASE( 100 )
!.....宏命令 setmat
            kmacr = lmacr( 2, kmacr )
            CALL UpdateStructure
            IF( nerrc .NE. 0 ) RETURN
          CASE( 101 )
!.....宏命令 time
            kmacr = lmacr( 2, kmacr )
            timec = para1
          CASE( 102 )
```

```
!.....宏命令 DOFWeight
                kmacr = lmacr( 2, kmacr )
                IF( npar1 .GE. 1 .AND. npar1 .LE. mdofn ) THEN
                  wdofs( npar1 ) = para2
                ELSE
                  nerrc = 565677
                END IF
              CASE DEFAULT
                WRITE( 12, 8000 ) ncomm
                nerrc = 45496
            END SELECT
            RETURN
8000        FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
            END

            SUBROUTINE Exec140
!.....
!           模块功能
!               本模块负责执行命令编号在 121-140 之间的各条宏命令.
!.....
            USE CtrlData
            USE MeshData
            USE CentData
            USE LoadData
            USE GlobData
            USE SolvData
            USE FrontData
            USE MacroData
            IMPLICIT DOUBLE PRECISION( a-h, o-z )

            ncomm = lmacr( 1, kmacr )
            para1 = pmacr( 1, kmacr )
            para2 = pmacr( 2, kmacr )
            para3 = pmacr( 3, kmacr )
            para4 = pmacr( 4, kmacr )
            para5 = pmacr( 5, kmacr )
            npar1 = pmacr( 1, kmacr ) + 0.5D0
            npar2 = pmacr( 2, kmacr ) + 0.5D0
            npar3 = pmacr( 3, kmacr ) + 0.5D0
            npar4 = pmacr( 4, kmacr ) + 0.5D0
            npar5 = pmacr( 5, kmacr ) + 0.5D0
            IF( ncomm .EQ. 121 ) THEN
!.....宏命令 backup
              kmacr = lmacr( 2, kmacr )
              CALL Backup
            ELSE IF( ncomm .EQ. 122 ) THEN
!.....宏命令 restore
```

```
            kmacr = lmacr( 2, kmacr )
            CALL Restore
          ELSE IF( ncomm .EQ. 123 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL SaveDisp
          ELSE IF( ncomm .EQ. 124 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL ReadDisp
          ELSE IF( ncomm .EQ. 125 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL SaveHistory()
          ELSE IF( ncomm .EQ. 126 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL ReadHistory()
          ELSE IF( ncomm .EQ. 127 ) THEN
            kmacr = lmacr( 2, kmacr )
            CALL ReadPrestress
          ELSE IF( ncomm .EQ. 128 ) THEN
!.....宏命令 savepres
            kmacr = lmacr( 2, kmacr )
            CALL UpdateStructure
            CALL SaveAsPrestress
          ELSE
            WRITE( 12, 8000 ) ncomm
            nerrc = 45496
          END IF
          RETURN
8000      FORMAT( //2x, '致命错误: 没有开发的宏命令', I5 )
          END

          SUBROUTINE AddCnt( iswth )
          USE CtrlData
          USE CentData
          USE SolvData
          USE MeshData
          IMPLICIT DOUBLE PRECISION( a-h, o-z )

          IF( iswth .EQ. 1 ) THEN
            DO istfp = 1, nstfp
              ipoin = lstfp( 1, istfp )
              idofn = lstfp( 2, istfp )
              ipoin = llnks(      ipoin )
              idofs = ipoin * mdofn + idofn - mdofn
              diagk( idofs ) = diagk( idofs ) + vstfp( istfp )
            END DO
          ELSE IF( iswth .EQ. 2 ) THEN
            DO imssp = 1, nmssp
              ipoin = lmssp( 1, imssp )
```

```
      idofn = lmssp( 2, imssp )
      ipoin = llnks(       ipoin )
      idofs = ipoin * mdofn + idofn - mdofn
      diagm( idofs ) = diagm( idofs ) + vmssp( imssp )
    END DO
  ELSE IF( iswth .EQ. 3 ) THEN
    DO idmpp = 1, ndmpp
      ipoin = ldmpp( 1, idmpp )
      idofn = ldmpp( 2, idmpp )
      ipoin = llnks(       ipoin )
      idofs = ipoin * mdofn + idofn - mdofn
      diagd( idofs ) = diagd( idofs ) + vdmpp( idmpp )
    END DO
  END IF
  RETURN
  END
```

附 4 ElmLib.f90 文本

```
      SUBROUTINE ElmLib( diags, vforc, iswth )
!.....
!     模块功能
!       形成总体矩阵: 刚度阵, 阻尼阵, 质量阵.
!.....
!     iswth 控制开关
!          == 0 读入材料信息          == 1 单元刚度阵
!          == 2 单元质量阵            == 3 单元阻尼阵
!          == 4 单元几何阵            == 5 内力等效节点力
!          == 6 体积力等效节点力      == 7 单元高斯点应力
!          ==-7 单元节点应力          == 8 单元信息
!          == 9 网格输出              ==10 节点反力
!.....
      USE CtrlData
      USE MeshData
      USE GlobData
      USE SolvData
      USE FrontData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION diags( ndofs ), vforc( ndofs )

      kcoun = 0                                          !标识进展
      IF( iswth .EQ. 4 ) isavc = lsavc( 2 )
      IF( iswth .LT. 4 ) isavc = lsavc( iswth )
      IF( isavc .EQ. 0 .AND. iswth .LE. 4 ) RETURN
!.....初始化工作变量和波阵.
      knume = 0
      npntw = 0
      smaxv = 0.0D0
      CALL InitInteger( lpntw, mpntw, 0 )
      CALL InitFloat( stifw, mstfw, 0.0D0 )
      CALL InitFloat( vforc, ndofs, 0.0D0 )
!.....根据记录信息组装矩阵.
      DO ircrd = 1, nrcrd
        inump = lrcrd( 1, ircrd )
        jnump = lrcrd( 2, ircrd )
        IF( isavc .EQ. 1 ) kdata = 0
        IF( isavc .LT. 0 ) kdata =-isavc - 1
        DO knump = inump, jnump
          kpoin = lbcks( 2, knump )
          kpntw = lbcks( 3, knump )
          kpnte = lpnte(      kpoin )
          DO ipnte = 1, kpnte
```

```
                knume = knume + 1
                kcoun = kcoun + 1
                kelem = lopts( knume )
                IF( necho .NE. 0 ) CALL ShowProcess( kcoun, nelem )
!.....形成当步波阵节点序号与节点总体编号对照表.
!.....形成元素节点序号与波阵节点序号对照表.
                DO inode = 1, mnode
                  lwave( inode ) = 0
                  jpoin = IABS( lnods( inode, kelem ) )
                  IF( jpoin .GT. 0 ) THEN
!.....当前节点为非零节点,查是否在波阵中.
                    jpoin = llnks( jpoin )
                    DO jpntw = 1, mpntw
                      IF( lwave( inode ) .EQ. 0 ) THEN
                        IF( jpoin .EQ. lpntw( jpntw ) )                    &
                        lwave( inode ) = jpntw
                      END IF
                    END DO
!.....如果当前节点不在波阵中,添加该节点.
                    IF( lwave( inode ) .EQ. 0 ) THEN
                      npntw = npntw + 1
                      jpntw = 1
                      DO WHILE( lwave( inode ) .EQ. 0 )
                        IF( lpntw( jpntw ) .EQ. 0 ) THEN
                          lpntw( jpntw ) = jpoin
                          lwave( inode ) = jpntw
                        ELSE
                          jpntw = jpntw + 1
                        END IF
                      END DO
                    END IF
                  END IF
                END DO
!.....形成元素矩阵并组装成总体矩阵.
                CALL ElmOpt( kelem, iswth )
                CALL slantb
                CALL Assemb( vforc, iswth, kelem )
                IF( nerrc .GT. 0 ) RETURN
              END DO
              IF( npntw .GT. 0 .AND. iswth .LE. 4 ) THEN
!.....存储总体矩阵对角元素.
                ndofw = npntw * mdofn
                DO idofn = 1, mdofn
                  idofw = ( kpntw - 1 ) * mdofn + idofn
                  idofs = ( kpoin - 1 ) * mdofn + idofn
                  istfw = idofw * ( idofw + 1 ) / 2
                  diags( idofs ) = stifw( istfw )
```

```
                IF( stifw( istfw ) .GT. smaxv )                    &
                    smaxv = stifw( istfw )
                stifw( istfw ) = 0.0D0
              END DO
!.....存储总体矩阵元素并收缩波阵
              IF( isavc .NE. 2 ) THEN
                DO idofn = 1, mdofn
                  locat = kdata
                  kdofw = ( kpntw - 1 ) * mdofn
                  idofw = kdofw + idofn
                  DO jpntw = 1, mpntw
                    IF( lpntw( jpntw ) .NE. 0 ) THEN
                      DO jdofn = 1, mdofn
                        jdofw = ( jpntw - 1 ) * mdofn + jdofn
                        IF( jdofw .LE. kdofw ) THEN
!.....当前行元素存储.

                          locat = locat + 1
                          istfw = idofw * ( idofw - 1 ) / 2 + jdofw
                          buffs( locat ) = stifw( istfw )
                          stifw( istfw ) = 0.0D0
                        ELSE IF( jdofw .GT. idofw ) THEN
!.....当前列元素存储.
                          locat = locat + 1
                          istfw = jdofw * ( jdofw - 1 ) / 2 + idofw
                          buffs( locat ) = stifw( istfw )
                          stifw( istfw ) = 0.0D0
                        END IF
                      END DO
                    END IF
                  END DO
                  kdata = kdata + ndofw - idofn
                END DO
              END IF
            END IF
!.....波阵.
            IF( npntw .GT. 0 ) THEN
              npntw = npntw - 1
              lpntw( kpntw ) = 0
            END IF
          END DO
          IF( isavc .EQ. 1 )                                       &
          CALL OptRec( buffs, ircrd, 2, iswth, nexch )
        END DO
        RETURN
        END

        SUBROUTINE ElmOpt( ielem, iswth )
```

```
!.....
!       模块功能
!           选择单元类型, 计算单元矩阵或应力.
!.....
        USE CtrlData
        USE MeshData
        USE ElmtData
        USE GlobData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        IF( iswth .EQ. 0 ) THEN
          imats = ielem
        ELSE IF( iswth .EQ. 8 ) THEN
          imats = ielem
          nnode = mnode
        ELSE IF( iswth .EQ. 9 ) THEN
          nnode = mnode
          iuses = luses( ielem )
          imats = lmats( ielem )
          DO inode = 1, mnode
            lnode( inode ) = lnods( inode, ielem )
          END DO
        ELSE
          CALL ElmPar( ielem )
          iuses = luses( ielem )
          IF( nerrc .GT. 0 ) RETURN
!.....设定单元信息.
          CALL ElmInf( ielem, iswth )
          IF( nerrc .GT. 0 ) RETURN
          imats = lmats( ielem )
        END IF
!.....调用相应的单元.
        itype = IABS( lprps( 1, imats ) )
        SELECT CASE( itype )
          CASE( 1 )
            CALL Elmt001( ielem, imats, iswth, iuses )
          CASE( 2 )
            CALL Elmt002( ielem, imats, iswth, iuses )
          CASE( 3 )
            CALL Elmt003( ielem, imats, iswth, iuses )
          CASE( 4 )
            CALL Elmt004( ielem, imats, iswth, iuses )
          CASE( 200 )
            CALL Elmt200( ielem, imats, iswth, iuses )
          CASE( 300 )
            CALL Elmt300( ielem, imats, iswth, iuses )
          CASE( 400 )
            CALL Elmt400( ielem, imats, iswth, iuses )
          CASE( 401 )
```

```
            CALL Elmt401( ielem, imats, iswth, iuses )
          CASE( 501 )
            CALL Elmt501( smaxv, ielem, imats, iswth, iuses )
          CASE DEFAULT
            nerrc = 24245
            WRITE( 12, 2000 ) itype
        END SELECT
2000    FORMAT( //2x, '错误: 无效的单元类型', I3// )
        RETURN
        END

        SUBROUTINE SlantB
!.....
!       模块功能
!           对单元刚度矩阵进行斜约束处理.
!.....
        USE CtrlData
        USE MeshData
        USE ElmtData
        IMPLICIT DOUBLE PRECISION( a-h, o-z )
        ALLOCATABLE tmatx(:,:), vmatx(:,:)
        IF( nslnt .LE. 0 ) RETURN
        ALLOCATE( tmatx( mdofn, mdofn ), vmatx( mdofn, mdofn ),                    &
                  STAT = nerrc )
        IF( nerrc .NE. 0 ) nerrc = 24435
        IF( nerrc .GT. 0 ) RETURN
        DO inode = 1, nnode
          ipoin = IABS( lnode( inode ) )
          IF( ipoin .GT. 0 ) THEN
            label = 0
            DO idofn = 1, mdofn
              DO jdofn = 1, mdofn
                tmatx( idofn, jdofn ) = 0.0D0
              END DO
              tmatx( idofn, idofn ) = 1.0D0
            END DO
            DO islnt = 1, nslnt
              jpoin = lslnt( 1, islnt )
              jdofn = lslnt( 2, islnt )
              IF( ipoin .EQ. jpoin ) THEN
                label = label + 1
                DO idofn = 1, mdofn
                  tmatx( jdofn, idofn ) = vslnt( idofn, islnt )
                END DO
              ELSE IF( jpoin .LT. 0 ) THEN
                idime =-jpoin
                IF( DABS( coord( idime, ipoin ) ) .LT. 1.0D-10 ) THEN
                  label = label + 1
```

```
          DO idofn = 1, mdofn
            tmatx( jdofn, idofn ) = vslnt( idofn, islnt )
          END DO
        END IF
      END IF
    END DO
    IF( label .GT. 0 ) THEN
      CALL Invers( tmatx, mdofn, mdofn, nerrc )
      IF( nerrc .GT. 0 ) RETURN
      DO jnode = 1, nnode
        DO idofn = 1, ndofn
          DO jdofn = 1, ndofn
            vmatx( idofn, jdofn ) = 0.0D0
            DO kdofn = 1, ndofn
              iloca = ( jnode - 1 ) * ndofn + idofn
              jloca = ( inode - 1 ) * ndofn + kdofn
              vmatx( idofn, jdofn ) = vmatx( idofn, jdofn ) +          &
              stife( iloca, jloca ) * tmatx( kdofn, jdofn )
            END DO
          END DO
        END DO
        DO idofn = 1, ndofn
          DO jdofn = 1, ndofn
            iloca = ( jnode - 1 ) * ndofn + idofn
            jloca = ( inode - 1 ) * ndofn + jdofn
            stife( iloca, jloca ) = vmatx( idofn, jdofn )
          END DO
        END DO
      END DO
      DO jnode = 1, nnode
        DO idofn = 1, ndofn
          DO jdofn = 1, ndofn
            vmatx( idofn, jdofn ) = 0.0D0
            DO kdofn = 1, ndofn
              iloca = ( inode - 1 ) * ndofn + kdofn
              jloca = ( jnode - 1 ) * ndofn + jdofn
              vmatx( idofn, jdofn ) = vmatx( idofn, jdofn ) +          &
              tmatx( kdofn, idofn ) * stife( iloca, jloca )
            END DO
          END DO
        END DO
        DO idofn = 1, ndofn
          DO jdofn = 1, ndofn
            iloca = ( inode - 1 ) * ndofn + idofn
            jloca = ( jnode - 1 ) * ndofn + jdofn
            stife( iloca, jloca ) = vmatx( idofn, jdofn )
          END DO
        END DO
```

```
            END DO
            DO idofn = 1, ndofn
              vmatx( idofn, 1 ) = 0.0D0
              DO jdofn = 1, ndofn
                iloca = ( inode - 1 ) * ndofn + jdofn
                vmatx( idofn,        1 ) = vmatx( idofn, 1 ) +                    &
                tmatx( jdofn, idofn ) * force( iloca      )
              END DO
            END DO
            DO idofn = 1, ndofn
              iloca = ( inode - 1 ) * ndofn + idofn
              force( iloca ) = vmatx( idofn, 1 )
            END DO
          END IF
        END IF
      END DO
      DEALLOCATE( vmatx, tmatx )
      RETURN
      END

      SUBROUTINE Assemb( vforc, iswth, ielem )
      USE CtrlData
      USE MeshData
      USE ElmtData
      USE FrontData
      IMPLICIT DOUBLE PRECISION( a-h, o-z )
      DIMENSION vforc( ndofs )

      imats = lmats( ielem )

      DO inode = 1, nnode
        ipntw = lwave( inode )
        IF( ipntw .NE. 0 ) THEN
          ipoin = lpntw( ipntw )
          DO idofn = 1, ndofn
            kdofn = lprps( idofn + 2, imats )
            IF( kdofn .GT. 0 ) THEN
              idofs = ( ipoin - 1 ) * mdofn + kdofn
              idofw = ( ipntw - 1 ) * mdofn + kdofn
              idofe = ( inode - 1 ) * ndofn + idofn
              vforc( idofs ) = vforc( idofs ) + force( idofe )
!.....选加节点载荷.
              IF( iswth .LE. 4 ) THEN
                DO jnode = 1, nnode
                  jpntw = lwave( jnode )
                  IF( jpntw .NE. 0 ) THEN
                    jpoin = lpntw( jpntw )
                    DO jdofn = 1, ndofn
```

```
                                kdofn = lprps( jdofn + 2, imats )
                                IF( kdofn .GT. 0 ) THEN
                                  jdofw = ( jpntw - 1 ) * mdofn + kdofn
                                  jdofe = ( jnode - 1 ) * ndofn + jdofn
                                  IF( jdofw .LE. idofw ) THEN
!.....选加单元刚度.
                                    istfw = ( idofw - 1 ) * idofw / 2 + jdofw
                                    stifw( istfw ) = stifw( istfw ) +                    &
                                    stife( idofe, jdofe )
                                  END IF
                                END IF
                              END DO
                            END IF
                          END DO
                        END IF
                      END IF
                    END DO
                  END IF
                END DO
                RETURN
                END

                SUBROUTINE ElmInf( ielem, iswth )
!.....
!       模块功能
!           从总体信息中获得单元信息.
!.....
                USE CtrlData
                USE MeshData
                USE ElmtData
                USE GlobData
                USE SolvData
                IMPLICIT DOUBLE PRECISION( a-h, o-z )
                ALLOCATABLE tmatx(:,:), vmatx(:)

                IF( nhstr .GT. 0 ) THEN
                  DO ihstr = 1, nhstr
                    histe( ihstr ) = histr( ihstr, ielem )
                  END DO
                END IF

                IF( nstrh .GT. 0 ) THEN
                  DO istrh = 1, nstrh
                    strhe( istrh ) = strsh( istrh, ielem )
                  END DO
                END IF
                DO inode = 1, nnode
                  lnode( inode ) = lnods( inode, ielem )
```

```
      ipoin   = IABS( lnods( inode, ielem ) )
      IF( ipoin .GT. 0 ) THEN
        tempe( inode ) = temps( ipoin )
        DO idime = 1, ndime
          coren( idime, inode ) = coord( idime, ipoin )
        END DO
        DO idofn = 1, ndofn
          idofs = ( ipoin - 1 ) * mdofn + idofn
          idofe = ( inode - 1 ) * ndofn + idofn
          dishe( idofe ) = disph( idofs, kload )
          dispe( idofe ) = disps( idofs, kload )
          dspie( idofe ) = dspis( idofs, kload )
          dispt( idofe ) = distm( idofs, kload )
          dishe( idofe ) = disph( idofs, kload )
          IF( iswth .EQ. 13 ) dispe( idofe ) = vslvs( idofs )
        END DO

        IF( ntypc .EQ. 2 ) THEN
          DO idofn = 1, ndofn
            idofs = ( ipoin - 1 ) * mdofn + idofn
            idofe = ( inode - 1 ) * ndofn + idofn
            accen( idofe ) = accel( idofs, kload )
            accet( idofe ) = acctm( idofs, kload )
            velen( idofe ) = veloc( idofs, kload )
            velet( idofe ) = veltm( idofs, kload )
          END DO
        END IF
      ELSE
        tempe( inode ) = 0.0D0
        DO idime = 1, ndime
          coren( idime, inode ) = 0.0D0
        END DO
        DO idofn = 1, ndofn
          idofe = ( inode - 1 ) * ndofn + idofn
          dispe( idofe ) = 0.0D0
          dspie( idofe ) = 0.0D0
          dishe( idofe ) = 0.0D0
          dispt( idofe ) = 0.0D0
          dishe( idofe ) = 0.0D0
          IF( ntypc .EQ. 2 ) THEN
            accen( idofe ) = 0.0D0
            velen( idofe ) = 0.0D0
            accet( idofe ) = 0.0D0
            velet( idofe ) = 0.0D0
          END IF
        END DO
      END IF
    END DO
```

```
IF( nslnt .LE. 0 ) RETURN
ALLOCATE( tmatx( mdofn, mdofn ), vmatx( mdofn ), STAT = ierro )
IF( ierro .NE. 0 ) nerrc = 35656
IF( nerrc .GT. 0 ) RETURN
DO inode = 1, nnode
  label = 0
  ipoin = IABS( lnode( inode ) )
  DO idofn = 1, mdofn
    DO jdofn = 1, mdofn
      tmatx( idofn, jdofn ) = 0.0D0
    END DO
    tmatx( idofn, idofn ) = 1.0D0
  END DO
  IF( ipoin .GT. 0 ) THEN
    DO islnt = 1, nslnt
      jpoin = lslnt( 1, islnt )
      jdofn = lslnt( 2, islnt )
      IF( ipoin .EQ. jpoin ) THEN
        label = label + 1
        DO idofn = 1, mdofn
          tmatx( jdofn, idofn ) = vslnt( idofn, islnt )
        END DO
      ELSE IF( jpoin .LT. 0 ) THEN
        idime =-ipoin
        IF( DABS( coord( idime, ipoin ) ) .LT. 1.0D-10 ) THEN
          label = label + 1
          DO idofn = 1, mdofn
            tmatx( jdofn, idofn ) = vslnt( idofn, islnt )
          END DO
        END IF
      END IF
    END DO
    IF( label .GT. 0 ) THEN
      CALL Invers( tmatx, mdofn, mdofn, nerrc )
      DO idofn = 1, ndofn
        vmatx( idofn ) = 0.0D0
        DO jdofn = 1, ndofn
          locat = ( inode - 1 ) * ndofn + jdofn
          vmatx( idofn            ) = vmatx( idofn ) +                    &
          tmatx( idofn, jdofn ) * dispe( locat )
        END DO
      END DO
      DO idofn = 1, ndofn
        locat = ( inode - 1 ) * ndofn + idofn
        dispe( locat ) = vmatx( idofn )
      END DO

      DO idofn = 1, ndofn
```

```
      vmatx( idofn ) = 0.0D0
      DO jdofn = 1, ndofn
        locat = ( inode - 1 ) * ndofn + jdofn
        vmatx( idofn          ) = vmatx( idofn ) +                    &
        tmatx( idofn, jdofn ) * dspie( locat )
      END DO
    END DO
    DO idofn = 1, ndofn
      locat = ( inode - 1 ) * ndofn + idofn
      dspie( locat ) = vmatx( idofn )
    END DO

    DO idofn = 1, ndofn
      vmatx( idofn ) = 0.0D0
      DO jdofn = 1, ndofn
        locat = ( inode - 1 ) * ndofn + jdofn
        vmatx( idofn          ) = vmatx( idofn ) +                    &
        tmatx( idofn, jdofn ) * dishe( locat )
      END DO
    END DO
    DO idofn = 1, ndofn
      locat = ( inode - 1 ) * ndofn + idofn
      dishe( locat ) = vmatx( idofn )
    END DO

    DO idofn = 1, ndofn
      vmatx( idofn ) = 0.0D0
      DO jdofn = 1, ndofn
        locat = ( inode - 1 ) * ndofn + jdofn
        vmatx( idofn          ) = vmatx( idofn ) +                    &
        tmatx( idofn, jdofn ) * dispt( locat )
      END DO
    END DO
    DO idofn = 1, ndofn
      locat = ( inode - 1 ) * ndofn + idofn
      dispt( locat ) = vmatx( idofn )
    END DO
    IF( ntypc .EQ. 2 ) THEN
      DO idofn = 1, ndofn
        vmatx( idofn ) = 0.0D0
        DO jdofn = 1, ndofn
          locat = ( inode - 1 ) * ndofn + jdofn
          vmatx( idofn          ) = vmatx( idofn ) +                  &
          tmatx( idofn, jdofn ) * velen( locat )
        END DO
      END DO
      DO idofn = 1, ndofn
        locat = ( inode - 1 ) * ndofn + idofn
```

```
            velen( locat ) = vmatx( idofn )
          END DO
          DO idofn = 1, ndofn
            vmatx( idofn ) = 0.0D0
            DO jdofn = 1, ndofn
              locat = ( inode - 1 ) * ndofn + jdofn
              vmatx( idofn          ) = vmatx( idofn ) +                    &
              tmatx( idofn, jdofn ) * accen( locat )
            END DO
          END DO
          DO idofn = 1, ndofn
            locat = ( inode - 1 ) * ndofn + idofn
            accen( locat ) = vmatx( idofn )
          END DO
          DO idofn = 1, ndofn
            vmatx( idofn ) = 0.0D0
            DO jdofn = 1, ndofn
              locat = ( inode - 1 ) * ndofn + jdofn
              vmatx( idofn          ) = vmatx( idofn ) +                    &
              tmatx( idofn, jdofn ) * velet( locat )
            END DO
          END DO
          DO idofn = 1, ndofn
            locat = ( inode - 1 ) * ndofn + idofn
            velet( locat ) = vmatx( idofn )
          END DO
          DO idofn = 1, ndofn
            vmatx( idofn ) = 0.0D0
            DO jdofn = 1, ndofn
              locat = ( inode - 1 ) * ndofn + jdofn
              vmatx( idofn          ) = vmatx( idofn ) +                    &
              tmatx( idofn, jdofn ) * accet( locat )
            END DO
          END DO
          DO idofn = 1, ndofn
            locat = ( inode - 1 ) * ndofn + idofn
            accet( locat ) = vmatx( idofn )
          END DO
        END IF
      END IF
    END IF
  END DO
  DEALLOCATE( tmatx, vmatx )
  RETURN
  END

  SUBROUTINE NodDof
  USE CtrlData
```

```
USE MeshData
USE GlobData
USE ElmtData
IMPLICIT DOUBLE PRECISION( a-h, o-z )

DO ipoin = 1, npoin
  ldofs( ipoin ) = 0
END DO
DO ielem = 1, nelem
  imats = lmats( ielem )
  itype = lprps( 1, imats )
  CALL ElmPar( ielem )
  DO inode = 1, nnode
    ipoin = lnods( inode, ielem )
    ipoin = IABS( ipoin )
    IF( ipoin .GT. 0 ) THEN
      IF( ndofn .GT. ldofs( ipoin ) ) ldofs( ipoin ) = ndofn
    END IF
  END DO
END DO
RETURN
END

SUBROUTINE ElmPar( ielem )
USE CtrlData
USE MeshData
USE ElmtData
IMPLICIT DOUBLE PRECISION( a-h, o-z )
iswth = 8
nnode = mnode
ndofn = mdofn
imats = lmats( ielem )
CALL ElmOpt( imats, iswth )
IF( nnode .GT. mnode ) nnode = mnode
IF( ndofn .GT. mdofn ) nerrc = 21435
IF( nerrc .GT. 0 ) RETURN
ndofe = nnode * ndofn
RETURN
END
```